

Technical Reference Series

Control Features of the IBM System/38



Ernst & Whinney

Technical Reference Series

Control Features of the IBM System/38

This table references illustrations reprinted by permission from International Business Machines (IBM) Corporation.

<u>Page Numbers</u>	<u>Publication Name</u>	<u>Copyright Years</u>
5,66	<u>IBM System/38 Introduction</u>	1978, 1980, 1982, 1983, 1984
33,37	<u>IBM System/38 Control Program Facility Programmer's Guide</u>	1979, 1980, 1981, 1982, 1983, 1984
43	<u>IBM System/38 Operator's Guide</u>	1980, 1981, 1982, 1983, 1984

Contents

INTRODUCTION	1
HARDWARE	3
CONTROL PROGRAM FACILITY	4
Control Language	4
Object Management.	7
Work (Job) Management.	8
Data Management.	9
Programmer Services.	16
System Operator Services	16
System Services.	18
Communications	19
UTILITIES.	20
Source Entry Utility (SEU)	20
Data File Utility (DFU).	22
Query Utility (QRY).	23
Screen Design Aid (SDA).	24
Display Information Facility (DIF)	24
SECURITY FEATURES.	25
User Profiles.	26
Group Profile Functions.	29
Security Officer User Profile.	31
System Authority	32
Object Ownership	34
Default Rights	34
Object Authority	36
ADDITIONAL SECURITY FEATURES	42
Initial Programs	42
Program Adopt Function	44
Libraries.	45
Logical Files.	46
Save and Restore	46
Console/Workstation Access	47
Passwords.	49
Workstation Inactivity	51
PROCEDURAL AND ADMINISTRATIVE CONTROLS	52
CONTROL CONCERNS AND AUDIT PROCEDURES.	56

CONTENTS

APPENDIX A--USING QUERY AS A COMPUTERIZED AUDIT TOOL 62

APPENDIX B--IBM-SUPPLIED PROFILES. 63

APPENDIX C--SELECTED IBM-SUPPLIED MENUS. 66

GLOSSARY 67

BIBLIOGRAPHY 73

Introduction

The IBM System/38 is a general purpose data processing system developed to manage on-line database environments. Many data processing systems require added programming efforts and software purchases to effectively maintain an on-line environment, but the necessary communications and database capabilities are included in the System/38 design. Although the System/38 was developed to support an on-line database environment, batch processing also is supported.

System/38 software is not compatible with the IBM System/34 or System/36.

This technical reference manual provides a source of information about the features and capabilities of the System/38 to be considered when identifying and evaluating controls in System/38 installations. This manual includes:

Hardware. A summary of hardware available for the System/38.

Control Program Facility. A description of the System/38 operating system and processing environment.

Utilities. A description of commonly used utilities available for the System/38 environment.

Security Features. A description of the basic security features available within the System/38 that can be used to provide control of the system.

Additional Security Features. Expands on the security features discussed in the previous chapter. Each section includes either an additional capability or a complexity within a previously discussed feature.

Procedural and Administrative Controls. A description of the procedural and administrative controls that, when implemented with the control features of the System/38, provide methods for controlling changes to program libraries and access to data files.

Control Concerns and Audit Procedures. A summary of the control considerations that should be addressed and suggested audit tests that may be performed in a System/38 environment.

INTRODUCTION

Also, appendices are included that address use of Query as an audit tool and that list IBM-supplied profiles and menus. A glossary of certain terms used in the manual and a bibliography of IBM manuals that provide more information about the System/38 also are included.

The contents of this manual are based on the System/38 operating system, Control Program Facility, Release 8, released in November 1986.

Hardware

The System/38 is available in several models. The primary difference between the models is the range of main and auxiliary storage available for each. Each system can include these components:

Main Storage
Up to 32 Megabytes.

Auxiliary Storage
Up to 14 Gigabytes.

System Console

Diskette Magazine Drive
The drive will hold up to two magazines (10 diskettes each) and three individual diskettes. Up to 24 megabytes of data may be written on two magazines before operator intervention is required.

Magnetic Tape Subsystem (Input/Output)

Workstation Controllers (Terminals)
Up to 256 workstations can be attached to the system.

Communication Attachments
Up to 12 communication lines can be attached to the system with multiple devices per line. Available communication software includes high-speed, local attachments with IBM System/34, System/36, System 370, 30XX, 43XX, and other System/38s; and high-speed, remote attachment with System/38 compatible systems or devices. Software is available to download data to personal computers.

Control Program Facility

The Control Program Facility (CPF) is the System/38 operating system. CPF resides in the System Library, QSYS. The major components of CPF are:

Control Language (CL)	Programmer Services
Object Management	System Operator Services
Work (Job) Management	System Services
Data Management	Communications

Exhibit 1 on page 5 groups CPF activities by major component. Each major component is a conceptual grouping of related functions within CPF. These major components do not represent the actual coding or organization of the operating system. Each major component included in the diagram is described in this chapter.

Control Language

The most common method of using CPF is through CL commands. The Control or Command Language is composed of easy to use instructions that initiate and direct the activities of the operating system. For example, CL is used to:

Create or delete a library, database, or user profile.

Restore a file.

Send a message to all workstations.

Execute a program.

Define a security profile.

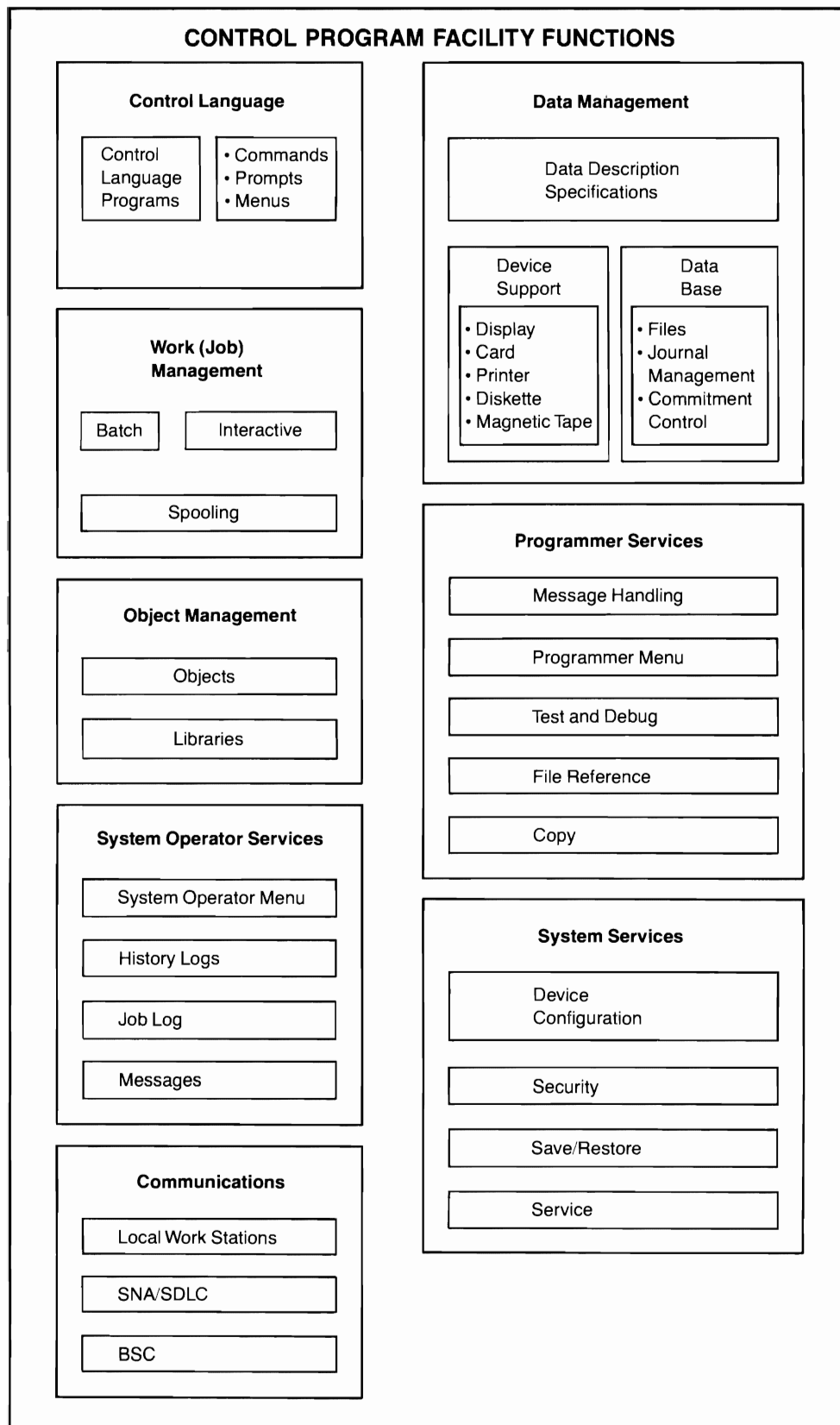
Back-up databases, programs, etc.

Print or display the history log.

Manage jobs (e.g., cancel, submit, or hold a job).

CONTROL PROGRAM FACILITY

Exhibit 1



CONTROL PROGRAM FACILITY

A list of all CL commands is available in the IBM System/38 Control Language Reference Manual. CL commands often perform sensitive functions. However, when the System/38 is installed, use of most commands is not restricted. Therefore, an installation should implement procedures that restrict access to libraries containing the CL commands, revoke access to sensitive commands, or restrict access to displays (screens) and menus that facilitate the use of commands. A detailed discussion of security options within the CPF operating system is included in the Security Features and Additional Security Features chapters of this manual. Each CL command consists of a command name and parameters specific to that command. Examples of CL commands and related parameters are:

```
CHGUSRPRF USRPRF(MJONES) PASSWORD(FLAG)
```

CHGUSRPRF is the "change user profile" CL command. In this example it changes the password for MJONES to FLAG. (User profiles and passwords are discussed on page 26.)

```
CNLJOB JOB(PAYABLE) SPLFILE(*YES) OPTION(*IMMED)
```

CNLJOB is the "cancel job" CL command. In this example the job named "PAYABLE" and the related items in the print spool (SPLFILE) are immediately (*IMMED) cancelled.

The first three letters of each command are the "verb" and the remainder are the "subject." In the examples, CHG (change) and CNL (cancel) are the verbs, USRPRF (user profile) and JOB (job) are the subjects. Because of the simple structure, CL commands are easy to interpret.

CL commands may be individually entered or included in a batch job stream. Additionally, a series of CL commands could be used in place of a source code program. For example, a CL program could be created to deactivate individual workstations on a specified schedule. A CL program is created by entering on-line the CL command series and then compiling the CL program. CL programs should be subject to the same access and review procedures applied to other programs.

CONTROL PROGRAM FACILITY

A user can enter a command using the command entry display or by using system prompts. The "command entry display" is a display screen with blank lines for commands to be entered on. There are no prompts for parameters on the command entry display. However, if the user selects the designated function key (usually CF4), CPF will provide a prompt with available parameters for the CL command requested. The prompt is a list of parameters with fill-in-the-blank spaces. The system generates and submits the actual command based on the parameters filled in.

Regardless of whether a command is entered from the command entry display or from a prompt, the command is edited for syntax errors when entered into the system. Help screens are available for detailed information about error messages.

Use of many CL commands is simplified by use of vendor, in-house, and IBM-supplied menus. CL commands can be invoked by selecting the menu options that represent the command and appropriate parameters.

Object Management

Object management is the use of CL commands to create, delete, and administer objects maintained on the system. Understanding the concept of an object is fundamental to an understanding of the System/38 environment. An object is the generic term for all "named items" stored on the System/38. Defining the term object is difficult but a review of items that are objects should clarify the definition. Objects include items found in many data processing environments and items specific to the System/38. Each of these items is an object on a System/38:

Libraries (1)

Programs (1)

Output queues (1)

Files (1)

Commands (2)

User profiles (2)

CONTROL PROGRAM FACILITY

Subsystem descriptions (2)

Job descriptions (2)

(1) Found in many data processing environments.

(2) Specific to the System/38.

Definitions of objects specific to the System/38 are included in the sections and chapters that follow.

CL commands are generally applied to an object. For example, the "restore library" CL command (RSTLIB) restores an object (i.e., a library). Certain CL commands can only be applied to certain types of objects. For instance, the "execute" CL command could not be applied to a file, however, it could be applied to a program that accesses a file.

System/38 security features are generally applied to objects. Access to an object can be restricted by user and by type of access (e.g., read, update, create). Access rights are discussed in the security chapters of this manual.

Work (Job) Management

The work management facilities provide the framework for managing the system and all work performed on the system. These facilities direct job input, execution, and output.

Work management facilities accept job submissions, obtain the resources required to run the job, and assign priorities according to predetermined criteria defined at the job or user level.

The System/38 supports both batch and interactive jobs. An interactive job is initiated when a user signs-on to a workstation. Each sign-on begins an interactive session and each session, sign-on to sign-off, is one job. Batch jobs may be initiated within an interactive job. Batch jobs are initiated by the use of CL commands. Batch jobs can be submitted at any workstation unless that right is specifically restricted as discussed in the security chapters of this manual.

CONTROL PROGRAM FACILITY

Jobs are executed within operating or work environments called subsystems. IBM supplies several predefined, work management subsystems. Examples are:

QINTER Supports the activity of interactive jobs.

QBATCH Supports batch jobs. QBATCH is started by a CL command. A batch job may be submitted even if QBATCH is not active; when QBATCH is started the job is waiting in the queue.

QSPL Supports the activities of the input and output devices, including spooling.

QPROG Supports the activities of programmers.

Additional subsystems can be created to support special processing needs. For example, an interactive application such as payroll could be maintained in a subsystem separate from QINTER. To strengthen internal control over payroll, access to the payroll subsystem could be limited to designated time frames. This is accomplished by entering CL commands at the system console to terminate or start the payroll subsystem at the discretion of management. This restricts the use of payroll while having no effect on the availability of interactive applications in the QINTER subsystem. Other methods of controlling access to programs and files are discussed in the security chapters of this manual.

Data Management

Data management is the use of CL commands to create, delete, and modify data objects. Within the System/38 environment, data management is a function of the operating system. This differs from data processing environments where databases are defined by a separate database management software package that must be purchased and interfaced with an operating system.

An important aspect of data management is the use of Data Description Specifications (DDS) to define the data fields within a given database (or database file) to the operating system. A detailed description of DDS is included later in this section.

CONTROL PROGRAM FACILITY

Within the System/38 environment, a file that is not a database file (a nondatabase file) has no DDS and consists only of a file name and the records in the file. The source code for each program accessing a nondatabase file must include the file description (e.g., the Data Division of a COBOL program must define the field names and attributes related to each field).

Use of nondatabase files generally is both inefficient and uncommon in the System/38 environment because, if properly implemented, the database facility of CPF provides more efficiency and control by:

- Reducing the number of program changes necessary if a record or field is modified (i.e., only the DDS needs to be changed).

- Providing uniform data definitions throughout the data processing environment.

For purposes of this manual, data management is divided into two sections. The first section describes how CPF uses DDS to manage database files. The second (1) describes how the Journal Management portion of the data management facility is used to provide a history of changes made to data files, and (2) defines the Commitment Control portion of the data management facility and describes how it can aid in determining that posting to data files provides consistent information to users of the system.

Data Description Specifications (DDS)

Data Description Specifications (DDS) describe the records within a particular database file. A DDS is part of the database file it describes and is entered as source code when the file is created. A DDS must be compiled, like all source code, before it is usable. For each data field in a record the DDS includes:

- Field name

- Field attributes, for example:

 - Data type (e.g., packed decimal, alphanumeric)

 - Field description

 - Number of character positions

There are three common types of database files that utilize DDS. These are field reference files, physical files, and logical files.

CONTROL PROGRAM FACILITY

Field Reference Files. A field reference file, similar to a data dictionary, can be created to provide uniformity in data field descriptions. A field reference file DDS includes information such as field name, number of characters, type of data (e.g., alphanumeric, packed decimal), acceptable range variables (e.g., an hourly pay rate field could be limited to a specified dollar amount), comments such as the field's mnemonics (i.e., a list of the various names used by different database files to describe the field), and the column heading to be used when the field is specified for print.

A field reference file DDS can provide consistent definitions of data throughout an installation while decreasing the time needed to create database files. An appropriate use of a field reference file DDS is to define fields where actual data records vary between database files but the field descriptions do not. For example, the zipcode, name, address, and telephone number fields would be identical for an account receivable database file and an employee payroll database file, but the actual data records in each file would be different (i.e., the accounts receivable customers would have different zip codes than the employees, but the field is still defined as a nine-digit numeric field). A field reference file does not contain data records. An example of how a field reference file DDS is used is included after the discussion on physical and logical files below.

Physical and Logical Files. A physical file contains data stored in fixed length records. A logical file lists and sequences selected fields contained in a physical file for use by a particular program or user. Both physical and logical files can be used or accessed by programs and users.

A logical file may exclude, reorder, rename, combine, or redefine fields that already exist in one or more physical files. A logical file appears to be a physical file to the program accessing it, however, a logical file does not actually contain any records.

Logical files can be used to provide access security. This example conceptually illustrates how a logical file can restrict access to data and how a field reference file, a physical file, and a logical file relate to each other.

CONTROL PROGRAM FACILITY

FILE-X INVENTORY DATABASE (a field reference file)

DDS

<u>Text</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Field Length</u>	<u>Comments</u>
Part Name	PRTNAM	A	8	ITMNAM
Part Number	PRTNUM	A	4	ITMNUM
Part Price	PRTPRC	N	6	
Customer Name	CUSNAM	A	30	
Customer Number	CUSNUM	A	4	
Address	ADDRES	A	30	
Zip Code	ZIPCOD	A	9	ZCODE
Phone Number	PHNNUM	A	10	CUSPHN, EMPPHN
Discount Allowed	DSCALD	N	2	

(Note: There are no records in field reference files.)

FILE-Y OUTSTANDING ORDER DATABASE (a physical file)

DDS

	<u>Text</u>	<u>Field Name</u>	<u>Data Type</u>	<u>Field Length</u>	<u>Comments</u>
(4)	Quantity Ordered	QTYORD	N	9	
(1)	REF(FILE-X)				
(2)	CUSNAM				
	CUSNUM				
(3)	ITMNAM	REFFLD(PRTNAM)			
	ITMNUM	REFFLD(PRTNUM)			
	DSCALD				

Records

	<u>CUSNAM</u>	<u>CUSNUM</u>	<u>ITNAM</u>	<u>ITNUM</u>	<u>DSCALD</u>	<u>QTYORD</u>
	TAC-Corp	3333	Part-1	111111	5	100
	BKB-Corp	4444	Part-2	222222	10	300
	WKV-Corp	5555	Part-1	111111	15	700

(6) FILE-Z ORDER ENTRY DATABASE (a logical file)

DDS

(5) REF(FILE-Y)
CUSNUM
ITMNUM
QTYORD

(Note: There are no records in logical files.)

CONTROL PROGRAM FACILITY

- (1) Indicates the fields (e.g., CUSNAM, CUSNUM, ITMNUM) are defined in FILE-X.
- (2) The field attributes for FILE-Ys CUSNAM are defined in a field by the same name in FILE-X, therefore, the attributes are not repeated here.
- (3) FILE-Ys field named ITMNUM is the same as FILE-Xs field named PRTNAM as designated by the field reference (REFFLD) parameter. Also, note ITMNUM is included in the comments as a mnemonic for PRTNAM in FILE-X.
- (4) Quantity ordered (QTYORD) was not defined in FILE-X (or any other field reference file not shown), therefore, the attributes for that field must be defined in FILE-Y.
- (5) Indicates the fields (e.g., CUSNUM, ITMNUM, QTYORD) are contained in FILE-Y. FILE-Z does not reference fields described in FILE-X because FILE-X contains no actual data for the fields defined.
- (6) In this example, order entry clerks could be defined as having access to FILE-Z. Order entry clerks would then only have access to the fields in FILE-Y that have been defined in FILE-Z (e.g., part number (ITMNUM), customer number (CUSNUM), quantity ordered (QTYORD)). Granting and revoking access to data files (objects) is discussed in the security chapters of this manual.

A source program using a database file need only reference the DDS--it does not need to describe the database fields (e.g., in a COBOL program, the Data Division includes the name of the file to be used by the program but does not describe the file). When the program is compiled, the operating system reads the file reference in the source program and copies the compiled DDS into the appropriate location in the compiled program based on the programming language used to write the source code (e.g., COBOL, RPG III).

If the description of a database file needs to be changed (e.g., to add or delete a field) only the DDS portion of the database file is modified. No modifications are made to programs that access the database file. However, the DDS and all programs that use the file must be recompiled after the change is made to the DDS.

CONTROL PROGRAM FACILITY

Level Check. Level check relates to the version of a DDS used to describe a file as compared to the version of a DDS copied into a program using the file. Each time a program is executed, the level number of the DDS is compared to the level number indicated in the program if *YES was specified in the level check parameter of the "create file" CL command. If the level identifier of the DDS accessed indicates it was compiled after the program, the program will not be executed. The default for this parameter is *YES. However, the default can be modified by the installation or changed at the discretion of the user when the "create file" CL command is used. The level identifier consists of a format and field description and is not related to dates or times.

If the level check parameter of the "create file" CL command is specified as *NO, the system will not automatically stop the program from executing against a DDS compiled after the program was compiled, which may result in processing errors. For example, assume a change to a DDS switches the character positions for the social security number and the zip code (both nine-digit numeric fields) within a payroll master file. If the application program updating the payroll master file was compiled before the change to the DDS was entered and compiled, each time the application program was used to change a zip code the new zip code would be written over the existing social security number.

Data Access Methods. The database environment of the System/38 allows simultaneous use of data by more than one user. However, while a record is retrieved for update by one user, other users may only read the record.

Data can be accessed using multiple access paths. When a file--physical or logical--is created, an access path is specified in the DDS. Access path specifications can be, but are not limited to, random, sequential, multiple key fields, ascending, and descending.

Device Data Management. Device data management facilitates the use of devices within the System/38 environment. A device description object defines each System/38 input/output device. A display device file describes the characteristics of System/38 input/output records. For example, a display device file could define the attributes of a menu (e.g., where data fields are to appear on a display terminal, what fields should be highlighted or blinking, and the input edits such as range variables and

CONTROL PROGRAM FACILITY

numeric checking that allow editing to be performed at the device before the data is transmitted to the file for posting).

Other Data Management Facilities

Journal Management. Journal management allows the user to record all changes made to a database. A journal can be designed for a specific purpose. For instance, one journal could be designed to provide an audit trail of transaction updates and another journal, with different attributes specified, could provide for file recovery. Journals can be used to:

- Review before and after images of database file changes.

- Summarize activity by:

- User
 - Time frame
 - Application
 - Workstation
 - File accessed

- Decrease the frequency of full database back-ups by only backing-up the changes.

- Facilitate recovery, if an object is damaged, by comparing the file to the changes recorded in the journal.

Installations may choose to dispose of on-line and/or hard copy of journals after a short time frame if large volumes of data are journaled. Therefore, the journals may not be available to external auditors for an entire period under audit.

Commitment Control. Commitment control is a feature within CPF that is often used when one transaction affects multiple records or files. To control the posting of related transactions, an installation may find it beneficial to place a database file under commitment control. The benefits derived by using commitment control are illustrated in these examples; using commitment control a user can:

- Enter all file updates associated with an individual transaction before any of the updates are actually written to the file.

- Remove changes made if the user detects an error in the transaction.

CONTROL PROGRAM FACILITY

Prevent other users from modifying a record related to a transaction in process until the entire transaction is posted.

Programmer Services

CPF includes many programmer assistance tools. Use of the tools is facilitated by the predefined Programmer Menu (see Appendix C). Programming aids available enable the programmer to:

- Use the Source Entry Utility to code source programs on-line.

- Compile programs from their workstations.

- Test and debug during program execution.

- Program from remote or local workstations.

- Utilize a test library for program development and testing.

- Copy objects.

- Access and modify certain message handling functions.

Also, a file referencing capability is available that allows a programmer to obtain, for example, this information:

- Database usage, organization, and definitions.

- All files using a specified record format.

- All logical files using a specified field in a physical file.

- All files used by a particular program.

Controls over programmer activities are discussed in the security chapters of this manual.

System Operator Services

System operator services consist of various operator assistance tools. CPF allows the system operator functions to be performed at any workstation. CL commands also may be entered from any workstation. However, the system console must be used to power on the system and to respond to the initial start-up display presented.

CONTROL PROGRAM FACILITY

System Operator Menu

The predefined System Operator Menu (included in Appendix C) provides traditional operator functions that simplify operator tasks. One option allows the operator to view the actual data in an output spool. This may be a control concern when sensitive data is in the output queue and access to output queues is not properly restricted.

System Logs

History Log. CPF logs and time stamps all messages to the system operator in the history file, QHST. The items logged include:

- Compiler usage.
- Utility usage.
- Job starts and ends, including abnormal endings.
- Invalid sign-on attempts, including which workstations were used.
- Invalid attempts to access objects, including which job made the attempt.

The history log can be regularly reviewed for unauthorized use of compilers, utilities, unauthorized access attempts, and any other unusual activities. Installations can easily review specific activities on a regular basis using the "display log" (DSPLOG) CL command. The command used to review password violations is shown on page 50.

An excerpt from a history log, which includes unauthorized password attempts, is shown.

```
SYSTEM LOG - QHST

MSGID      SEV  MSG TYPE
CPA5323  99  INQUIRY      Ribbon check on device QSYSPRT (C R 1-9999) ERR 0018-5436-68715487C2C0N0CE.
          QSYSPRT  QSPLJOB  064275  11/13/86 10:39:46
          VALID RPY
          R
          QCONSOLE  QSYSPRT  064437  11/13/86 10:39:46
CPA5323  99  INQUIRY      Ribbon check on device QSYSPRT (C R 1-9999) FRP 0018-5436-68715C388FC0N0CC.
          QSYSPRT  QSPLJOB  064275  11/13/86 10:39:55
CPA3394  99  INQUIRY      Load form type *ESTD* dev QSYSPRT2 wtr QSYSPRT2 (H C G I R).
          QSYSPRT2  QSPLJOB  064270  11/13/86 10:41:15
          VALID RPY
          G
          QCONSOLE  QSYSPRT  064437  11/13/86 10:41:15
CPA5316  99  INQUIRY      Verify alignment on device QSYSPRT2 (I G R N C).
          QSYSPRT2  QSPLJOB  064270  11/13/86 10:41:23
CPA5323  99  INQUIRY      Ribbon check on device QSYSPRT (C R 1-9999) FRP 0018-5436-68715C388FC0N0CC.
          QSYSPRT  QSPLJOB  064275  11/13/86 10:41:41
          VALID RPY
          R
          QCONSOLE  QSYSPRT  064437  11/13/86 10:41:41
CPA5316  99  INQUIRY      Verify alignment on device QSYSPRT2 (I G R N C).
          QSYSPRT2  QSPLJOB  064270  11/13/86 10:41:47
          VALID RPY
          I
          QCONSOLE  QSYSPRT  064437  11/13/86 10:41:47
CPA5313  99  INQUIRY      Forms jammed on device QSYSPRT2 (C R 1-9999) ERR 0058-5437-68715CA948N00T1.
          QSYSPRT2  QSPLJOB  064270  11/13/86 10:41:51
CPA5313  99  INQUIRY      Forms jammed on device QSYSPRT2 (C R 1-9999) FRP 0054-5437-68715CA948N00T1.
          QSYSPRT2  QSPLJOB  064270  11/13/86 10:42:23
          VALID RPY
          R
          QCONSOLE  QSYSPRT  064437  11/13/86 10:42:23
CPF2234  30  INFO      Invalid password entered from device PROG4.
          QINTER  QSYS  064257  11/13/86 10:45:26
CPF2234  30  INFO      Invalid password entered from device PROG4.
          QINTER  QSYS  064257  11/13/86 10:45:31
CPF2234  30  INFO      Invalid password entered from device PROG4.
          QINTER  QSYS  064257  11/13/86 10:45:36
CPF2234  30  INFO      Invalid password entered from device PROG4.
          QINTER  QSYS  064257  11/13/86 10:45:41
CPF2234  30  INFO      Invalid password entered from device PROG4.
          QINTER  QSYS  064257  11/13/86 10:45:47
```

CONTROL PROGRAM FACILITY

Job Logs. The system also maintains a "job log" that records the detail of activities that occurred within a batch or interactive job. Job logs are designed to record more detail than the history log. However, the level of detail recorded can range from "none" to "all messages sent." When a job description is created the level of detail to be logged each time the job is executed is specified in the LOG parameter of the CL command "create job description" (CRTJOBDD).

The job log can be used to review items such as error messages, system messages to the user, menu options selected, and job requests (e.g., a request to access a file). A cumulative file of all job logs is not maintained. When a job is completed each job log may be printed or deleted at the user's discretion. Accordingly, it is difficult to account for all job logs, therefore the job log has limited audit use.

Messages

CPF processes both automatic system responses and manually entered operator responses. An operator may send a message from any terminal to any other terminal including the system console.

System Services

System service functions assist an installation in managing the system operation. Included in these functions are:

Device configuration. Configuration commands are entered for optional devices added to the system. Standard devices such as the diskette magazine need not be configured by the installation; however, the IBM configuration of these devices may be modified if desired. Input/output devices are configured by use of the "create device" (CRTDEVDD) CL command. The command parameters specify device address, type of device, etc.

Security. CPF includes several security features; however, implementation is optional and at the discretion of management at each installation. CPF security features are discussed in detail in the Security Features and Additional Security Features chapters of this manual.

Save/Restore. The save function can save all objects on the system, selected objects, or a group of objects by writing them to a diskette magazine drive or magnetic tape device.

CONTROL PROGRAM FACILITY

The restore function allows the restoring of objects and libraries that have been saved off-line. The save/restore history information for each object records when and where each object was saved and when the object was last restored. The information can be displayed using CL commands and may be used to determine that objects are not inadvertently restored from an outdated copy of the object. Security features related to save/restore are discussed on page 46.

Service. CPF enables service representatives to perform most service functions concurrently with normal data processing operations. With concurrent maintenance support, the system operator and service representative can share the use of the system console if desired. The service facilities of CPF provide support for handling problems related to CPF, workstations, and the system unit. Hardware and software errors are recorded in two logs--machine and service. The error information is retrieved by using the "list error" command for the machine error log and the "display log" command for the service log.

Communications

The design of CPF allows for a wide variety of communication capabilities. The System/38 can function as a workstation with a host system, a host for multiple workstations, and on an equal level with other System/38s. Once the system has been accessed through a communication link, access to system objects is authorized or denied using the same security conventions as all other accesses. Security features of the system are described in subsequent chapters.

Utilities

IBM's Interactive Data Base Utilities (IDU) is a software package consisting of several data handling utilities available for the System/38. IDU is not included in the base purchase price of the System/38, however, virtually all installations purchase it. IDU consists of these utility programs:

Source Entry Utility	(SEU)
Data File Utility	(DFU)
Query Utility	(QRY)
Screen Design Aid	(SDA)

Display Information Facility (DIF), is an additional utility program available for the System/38. IDU and DIF are the most commonly used utilities that may affect the control environment. Methods of controlling access to utilities are discussed in the security chapters of this manual.

Source Entry Utility (SEU)

SEU allows users to create and maintain source code on-line. SEU may be used to create and maintain Control Language (CL) source code programs; Data Description Specifications (DDS); and RPG III, COBOL, and BASIC source code.

Statements are edited for syntax as entered. CL parameter prompting and CL commands from existing menus may be used when entering CL source statements using SEU.

SEU can be used to perform these functions:

Add, delete, change, move, and copy source code.

Scroll through a source member.

Locate a specified string of characters.

SEU also records all changes, additions, and deletions made to a source file and logs the date of creation or latest change date in the far right hand columns of each source statement line. The date logged is the date maintained by the operating system (i.e., if the operating system start-up date entered at system initialization is in error, the SEU log dates also will be in error). Although the date logged can be viewed on-line, the field is controlled by the operating system and cannot be edited by users.

UTILITIES

An excerpt of a RPG III source file is shown.

```

5714UT1 R07M00 850913          SEU SOURCE LISTING          17:37:16          PAGE    3
SOURCE FILE:      CMMRPG.CMMSRC          MEMBER:    CMMRDLT
SEQNBR*... 1 ... 2 ... 3 ... 4 ... 5 ... 6 ... 9 ... 0

10700      C          *IN02      DDWEO' 0*                      09/13/85
10800      C          FDBDIS      IFLE DATEE                      09/13/85
10900      C          FDBDIS      IFGE DATEE                      09/13/85
11000      C          ADD      1          DLTFIN  90              09/13/85
11100      C          DELETFBDBREC                      09/13/85
11200      C          END                      09/13/85
11300      C          ELSE                      09/13/85
11400      C          MOVE  *1*          *IN02                  09/13/85
11500      C          END                      09/13/85
11600      C          *IN02      IFEO' 0*                      09/13/85
11700      C          READ FBDBREC                      02          11/01/85
11800      C          END                      09/13/85
11900      C          END                      11/19/85
12000      *                      09/13/85
12100      *                      09/13/85
12200      C          END                      09/13/85
12300      *                      09/13/85
12400      *                      09/13/85
12500      C          EXSR NEXTDA                      11/19/85
12600      C          DLTMED      ADD      DLTFIN      TSTDLT  90  10/11/85
12700      C          DBRECS      SUB      TSTDLT      DBREC      10/30/85
12800      C          Z-ADD1      DBDEL                      10/31/85
12900      C          TSTDLT      IFGT  0                      10/11/85
13000      C          UPDATESYSUPD                      10/11/85
13100      C          DBRECS      DIV      TSTDLT      DBPRE      54H  10/30/85
13200      C          DBPRE      MULT  100      DBPCT      52    10/30/85
13300      C          END                      10/11/85
13400      C          END                      11/19/85

```

From the excerpt it can be determined that line 10700 was entered or last changed on 9/13/85, line 11700 was entered or last changed on 11/1/85, etc.

The SEU date log can be a useful tool for comparing actual program changes made to those changes authorized. Additionally, auditors can use the SEU date log to benchmark the testing of an application. The compile date (i.e., change or creation date) can be used to determine if the source code reviewed correlates with the executable program in production.

The SEU date log can be used to compare the program changes actually made to changes authorized. For example, the only change authorized for an interest calculation program after 9/30/X7 is a change to the "maximum allowable interest rate" entered on 12/2/X7. The source code could be reviewed for lines of code changed after 9/30/X7. The only date after 9/30/X7 should be 12/2/X7 on the source code line with the interest rate. Since only the latest change date for each line is shown, the reviewer would be unable to determine if other changes to this line were made between 9/30/X7 and 12/2/X7.

UTILITIES

Auditors can use the SEU date log to benchmark audit procedures performed on a particular accounting application. Using procedures similar to those described above, an auditor can review the SEU date logs for an application and determine which lines of code were modified, if any, after the audit procedures were performed. Changes made can be evaluated for significance (e.g., a report title change compared to the rewrite of the warranty reserve calculation) and additional audit procedures performed as appropriate.

After the source code changes are identified using the SEU date logs, the "display object description" (DSPOBJD) CL command can be used to obtain the date the source code was compiled. The create or change date shown for the compiled program should be after the latest SEU date log in the related source code program.

The compilation date (date created or date changed) obtained using the DSPOBJD command also can be reviewed to verify that no changes were made to a program after a specified benchmark date.

Access to SEU should be restricted to authorized personnel. A detailed discussion of System/38 security features and control concerns is included in subsequent chapters.

Data File Utility (DFU)

Through a series of menus and prompts, DFU allows the user to quickly create programs that will add to, delete from, create, read, move, or modify data files without using other programming procedures. DFU is simple to use and requires no programming experience.

Database fields used in DFU programs are defined in Data Description Specifications (DDS), therefore the user needs only to name the file to be used. DFU may be used on both logical and physical files. Use of DFU to modify data files circumvents programming controls (e.g., editing of data before posting) usually present in traditional application programs.

Although use is not mandatory, an unsophisticated and optional audit control prompt permits the creator of a DFU program to specify YES or NO to these inquiries:

Are additions and deletions of records allowed?

UTILITIES

Are changes to records allowed?

Do you want to print a log of additions, changes, or deletions?

Once created, the DFU program is executed in accordance with the creator's responses to these questions regardless of who executes the program.

If the audit control logging option is specified, a detailed record of data file changes made by the DFU program is maintained. Because the audit control prompts within a DFU program can be modified, access to the DFU program must be restricted. The population of DFU changes is difficult to verify because usage of the audit control prompt is optional, therefore the logs of additions, changes, and deletions has limited audit use.

DFU can be used to perform certain computer-assisted audit procedures (e.g., footing of fields, extracting information); however, "Query" is a better audit tool as discussed in Appendix A.

Query Utility (QRY)

Query simplifies database inquiry procedures. Query allows users to interactively specify criteria for the extraction, summarization, and presentation of information from a database. The information can be formatted into a report that can be displayed on a screen or printed.

Query uses the data field definitions maintained in a database's DDS; therefore, users need only supply the name of the file to be used. Because Query applications are created using menus, prompts, and help texts, no programming experience is required.

Query allows limited computations such as totalling fields, averaging fields, and obtaining record counts. Records can be extracted based on specified criteria. Query can be a valuable audit tool for accessing information held in a database. See Appendix A for a discussion of audit uses.

UTILITIES

Screen Design Aid (SDA)

SDA allows the user to interactively design, create, maintain, and test workstation display formats and menus for applications. SDA generates DDS for the displays and the menus designed and creates the CL source programs that support the menus. A security concern related to using SDA-designed menus is discussed on page 43.

Display Information Facility (DIF)

DIF is a programmer productivity aid. DIF programs can be used to retrieve and display data file information on a workstation screen. Using a DIF program a user may inquire, add to, delete from, modify, search, and list data files. The search functions include search by alphabetic key, character strings, or by scrolling through sequential records.

Writing DIF programs requires minimal programming skill; on-line assistance is available if needed.

Security Features

The System/38 provides a set of security features that can be used to control access to objects. Security features are standard within the System/38 operating system; however, their implementation is optional and at the discretion of management at each installation.

System/38 security features provide an infinite number of possible control combinations. These combinations range from not implementing any security features to implementing all available features. An installation should outline its security goals and appropriate procedures should be defined and implemented to achieve those goals. The procedures may rely heavily on controls (e.g., passwords, edits) within the individual accounting applications or may use the security features available within the System/38. In many instances, an installation may use a combination of both. Application controls are not addressed in this manual but should be considered in an evaluation of the system of internal controls related to significant accounting applications.

There are two security chapters in this manual--Security Features and Additional Security Features. The first three sections of the first security chapter--User Profiles, Group Profile Functions, and Security Officer User Profile--discuss how individuals (or groups of individuals) are represented to the operating system. The last four sections--System Authority, Object Ownership, Default Rights, and Object Authority--discuss how user profiles and CL commands are used to specify which access rights (e.g., read, add, delete) individuals have to objects.

Additional capabilities or complexities of the basic components discussed in the Security Features chapter are explained in the second security chapter. It includes how Initial Programs, libraries, logical files, passwords, and other features can be used to provide installation security.

The security sections of this manual should provide a basis for understanding the individual security features available in the System/38 operating system. However, since individual features may be combined to form many appropriate security combinations, each installation must be evaluated separately to determine if that environment is adequately controlled.

SECURITY FEATURES

User Profiles

A user's ability to access objects is allowed or denied based on information designated in a "user profile." A user profile contains information about a user, or group of users, and the objects that the profile can access. For purposes of System/38 security, a "user" is anyone using the system including both nondata processing personnel (e.g., accounting clerks, terminal operators) and data processing personnel (e.g., programmers, system operators).

Each user profile is an object. Key parameters included in a user profile are:

<u>Parameter Name</u>	<u>Parameter Description</u>
USRPRF	User Profile--The name of this user profile object and the user ID.
PASSWORD	Password--The code needed to respond to the "Enter password" display when shown on the screen.
MAXSTG	Maximum Auxiliary Storage--The maximum space allotted for objects "owned" by this user. "Ownership" is discussed on page 34.
PTYLMT	Highest Scheduling Priority--Highest spooling priority available to this user. Options available are one through nine, with one being highest.
GRPPRF	Group Profile--The name of the group profile this user is associated with. This may be designated as *NONE. Group profiles are discussed on page 29.
INLPGM	Initial Program--The default program called when the user signs-on to the system--usually a menu. This may be designated as *NONE. Initial Programs are discussed on page 42.
OWNER	Owner--This defines whether the default "owner" of objects created by this profile is to be the user profile or the group profile designated in the GRPPRF parameter (if any).

SECURITY FEATURES

GRPAUT Group Authority--The default access granted to other members in this user's group profile when objects are created by this user.

SPCAUT Special Authority--Special Authorities available are:

System Save Rights--SPCAUT(*SAVSYS). User can save, restore, and free all objects on the system. Freeing an object allows the system to use the space the object maintained; however, freeing is different than deleting because a freed object's description remains on the system and the object can be restored to the system without being redefined. If an object is deleted, the object description also is deleted.

Job Control Rights--SPCAUT(*JOBCTL). Authorizes the user to hold, release, cancel, and change the input, output, and processing status of any job that can be controlled by the system operator. Users also may power down or initialize the system, terminate the operating system, and/or vary on or off (deactivate or activate) any subsystem (e.g., the batch or interactive subsystem). System initializations must be performed at the system console by a user with SPCAUT(*JOBCTL).

Administrator's Rights--SPCAUT(*ADMIN). This relates to office management software available for the System/38 called, OFFICE/38. OFFICE/38 offers wordprocessing, message handling, and other office administration-related applications. As this software is optional and generally has little impact on significant accounting applications, administrator's rights are not discussed in this manual.

For most users this parameter should be designated as *NONE because these functions relate to systemwide operational authorities not specific to individuals or objects. Generally these functions are the responsibility of operations personnel.

SECURITY FEATURES

The information that follows also is included in the user profile. However, this information is not entered as a parameter.

Owned Objects. List of objects owned by this user profile. This includes objects owned due to a transfer of ownership by a previous owner or the security officer, and objects created by this user profile. Ownership of objects is discussed on page 34.

Authorized Access. List of objects owned by another user profile but a specific level of access to the object has been explicitly granted to this user profile. The list does not include public authority access allowed all users (including this user). Objects authorized to the public are not detailed in the profile. Object authority is discussed on page 36.

When a user creates an object, or a user is granted authority to an object, that information is posted to the user profile (see discussion on page 34).

Usually, a user signs-on to the system by entering a user profile object name (i.e., user ID) and a password. The password is not displayed on the screen when entered. The system verifies the password is valid for the user ID before allowing access to the system. The information in the user profile is then used to determine if access to objects will be allowed when access is attempted. Additional password discussion is included on page 49.

Certain IBM-supplied user profiles are present when the system is installed. Their functions and object names are:

Security Officer	(QSECOFR)
Programmer	(QPGMR)
Workstation User	(QUSER)
System Operator	(QSYSOPR)

These IBM service representative user profiles also are supplied:

Programming Service Representative	(QPSR)
Customer Engineer	(QCE)

The default parameters for each IBM-supplied user profile are included in Appendix B.

Each IBM-supplied user profile is installed with a separate password. However, all System/38s are installed with the same standard passwords that are clearly printed in the System/38

SECURITY FEATURES

system documentation manuals. Therefore, installations should change all predefined user profile passwords whether or not the user profile will be used.

The system allows multiple users to share a user profile. Users sign-on using the same user ID and password and are represented to the system as one user profile. For example, the user profile "USR" with the password "WINDOW" may be used simultaneously by as many users as there are workstations. Each user profile may be assigned only one password. Therefore, to require a confidential password for each user, and individual accountability for activity, separate user profiles must be created for each user.

Group Profile Functions

A group profile allows several individual users access to the same objects without repeating the authorizations in each user profile. Group profiles conserve system storage space and decrease input time when granting or revoking object authorization to a group of users.

A group profile can be any existing user profile or a user profile set up specifically to be a group profile. Functionally, and to the system, there is no difference between the two. Any user profile becomes a group profile when its object name is placed in the group profile parameter of another user profile.

For example, assume these parameters are excerpts from four user profiles. Also assume public authority (see page 34) is *NONE for the files and programs listed.

User Profile Name: User-A
Group Profile: *NONE
Authorized Objects:
 File1
 Program1

User Profile Name: User-B
Group Profile: User-A
Authorized Objects:
 File2
 Program2

User Profile Name: User-C
Group Profile: User-A
Authorized Objects:
 File3
 Program3

User Profile Name: User-D
Group Profile: User-A
Authorized Objects:
 File4
 Program4

SECURITY FEATURES

In this example:

User-B can access File1, File2, Program1, and Program2.
User-C can access File1, File3, Program1, and Program3.
User-D can access File1, File4, Program1, and Program4.

User-A can only access File1 and Program1. Anything created by or authorized to User-A can be accessed by User-A, User-B, User-C, and User-D.

Objects authorized to group members are not authorized to other group members unless each has been granted explicit authorization also. In the example, User-B cannot access File3 and Program3 because access has not been granted to User-B or User-A.

CPF allows a user profile to be created without a password. A user profile cannot be used if there is no password for sign-on. This allows a user profile to be created specifically for use as a group profile.

An existing user profile may not be appropriate as a group profile. The existing profile may contain object authorizations that should not be granted to all group members or the existing owner could create an object that should not be available to the group. If User-A created a confidential file, User-B, User-C, and User-D could not be denied access to the file.

If an existing user profile is used as a group profile and that user leaves the group due to termination, promotion, or transfer, numerous changes to several user profiles may be necessary to re-establish appropriate authorizations. For example, if User-A transfers to another department and should not have access to File1 and Program1, access to both objects should be revoked from User-A. The group profile parameter for User-B, User-C, and User-D should be changed because User-A's new authorizations should not be authorized to User-B, User-C, and User-D. Also, access to File1 and Program1 must be granted to User-B, User-C, and User-D either individually or through a new group profile. In this example only four files and four programs are used. However, in many situations several hundred files and programs may be affected.

Each user signs-on to the system using his or her own user ID and password. Therefore, individual user accountability is maintained when using group profiles except when multiple users share the same user ID and password (e.g., the entire accounts receivable department is assigned the User-B user profile).

SECURITY FEATURES

The security officer user profile can obtain a list of members in each group profile (see page 32). Group profiles can be utilized in a variety of combinations that do not compromise system security. An understanding of the basic group profile concepts in this section and a review of how an installation uses group profiles should provide the auditor with the necessary information to evaluate the propriety of its usage by the installation.

Security Officer User Profile

The security officer user profile (QSECOFR) has complete access to all aspects of the system and is responsible for all security procedures. The first responsibility of the security officer should be to change the password for the QSECOFR user profile from the standard IBM-assigned password.

By system design, the security officer user profile cannot be deleted nor can there be more than one security officer user profile. However, the security officer profile can be designated as a group profile parameter in any other user profile. Therefore, any user profile designated as "GRPPRF(QSECOFR)" can access all aspects of the system.

The security officer's capabilities include:

- Creating user profiles for users or assigning users one of the predefined IBM user profiles.

- Granting or revoking a user's authority to use specific system functions (CL commands), application programs, and other objects by modifying the user profile or the affected objects.

- Revoking a user's authority to sign-on to the system by changing the password or deleting the user profile from the system.

- Displaying a list of user names and passwords.

- Displaying the contents of a user's profile.

- Displaying the object authority information for all objects.

There are five Control Language (CL) commands that can only be used by the security officer (and those user profiles with QSECOFR designated as their group profile). Access to these commands cannot be granted to other user profiles:

SECURITY FEATURES

<u>Command</u>	<u>Purpose</u>
CHGUSRPRF	Change user profile
CRTUSRPRF	Create user profile
DSPAUTUSR	Display authorized users
DLTUSRPRF	Delete user profile
DSPUSRPWD	Display user password

Even though use of these commands cannot be granted to other users, a diagnostic message is sent if there are attempts to grant access to these commands to any user. For example, if a user profile, designated as QSECOFR in the group profile parameter, attempted to grant an accounts payable user profile use of the CHGUSRPRF command (1) the attempt would be denied and (2) the attempt would be logged.

DSPAUTUSR will display or print an alphabetical list of user profiles by profile name. User profiles designated as group profiles are indicated by an "X" on the list. A list of the members of each group profile can be obtained by selecting the *GRPMBR option of the display user profile CL command (DSPUSRPRF).

Passwords can be displayed using DSPUSRPWD if "0" was specified in the "QSIGNLVL" system value at initialization. If "1" was specified, passwords are encrypted and not available even to the security officer (see discussion on page 50).

System Authority

System Authority is composed of "Special Authority," the authorizations needed to perform certain functions that affect the entire system and "Object Authority," the authorizations needed to perform certain functions that affect only specific objects. Special Authority is specified in a user's profile (see page 27) and Object Authority is granted to the objects (see page 38) and then posted to the user profile. An outline of these areas and their subcomponents includes:

- I. Special Authority. Relates to systemwide authorizations.
 - A. System Save Rights. User profile can save, restore, and free all objects on the system. The security-related aspects of save and restore are discussed on page 46.
 - B. Job Control Rights. Authorizes the user profile to hold, release, cancel, and change the input, output, and processing status of any job that can be controlled by

SECURITY FEATURES

the system operator. Users also may power down or initialize the system and terminate the operating system and the subsystems such as the batch or interactive subsystem.

C. Administrator's Rights. Refer to comments on page 27.

II. Object Authority. Relates to object-specific authorizations.

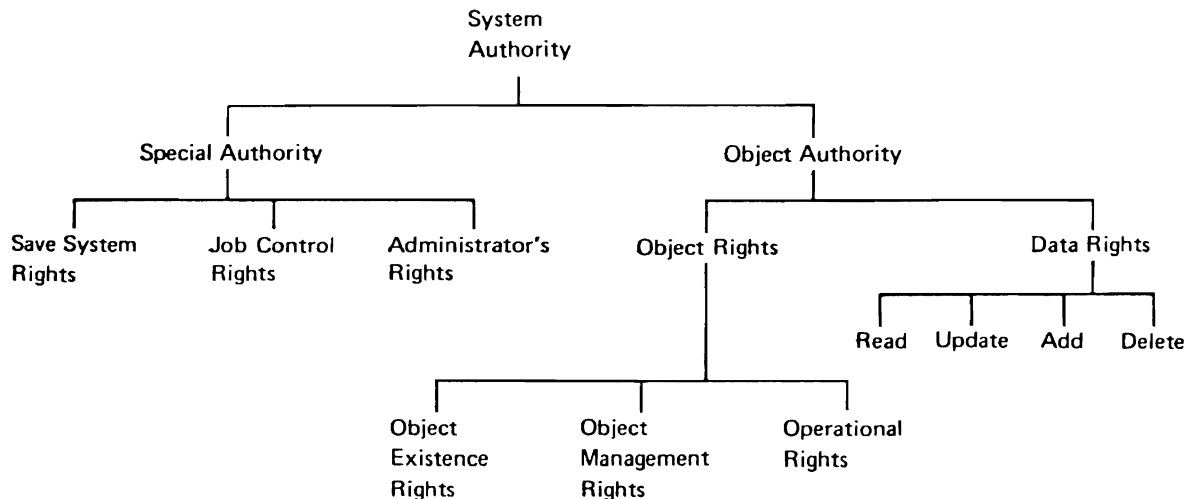
A. Object Rights.

1. Object Existence Rights. User can delete, save, restore, and transfer ownership of the object.
2. Object Management Rights. User can move, rename, grant authority to, revoke authority from, and modify the attributes of the object.
3. Operational Rights. User can use (e.g., execute) the object and view the object's description.

B. Data Rights.

1. Read
2. Update
3. Add
4. Delete

A better understanding of System/38 security may be obtained by referring to the definitions when the terms are encountered in subsequent sections. This illustration summarizes the System/38 authorities and rights:



SECURITY FEATURES

Throughout this manual, and the IBM documentation listed in the Bibliography, the terms "authorization" and "access" are used to indicate whether or not a user profile has been given the "rights" to perform a given procedure.

Object Ownership

All objects have only one owner. When a user profile creates an object, that user profile becomes the "owner" of the object. Unless the object's ownership is transferred to another user profile, the creating user profile remains the owner until the object is deleted from the system.

The owner has complete control over owned objects. The owner has all Object Rights and all Data Rights. The owner may grant rights to other users or may transfer ownership to another owner. If ownership is transferred, the new owner may revoke or decrease the rights of the old owner, but unless that occurs, the old owner retains all previous rights (i.e., the same rights as the new owner).

Default Rights

When an object is created, the creating user may specify the default level of access allowed to the public (all user profiles). This default level is defined in the PUBAUT parameter of the object description. The options include:

- *ALL All user profiles have authority as if they were the owner.
- *NONE Only the owner, the security officer, and those explicitly granted authority have access.
- *NORMAL All users have certain authorizations to the objects.

When PUBAUT is defined as *NORMAL, default authorizations vary by object type. This example depicts common object types and the authorizations (as indicated by an "X") allowed when public authority is designated as *NORMAL:

SECURITY FEATURES

	<u>Object Rights</u>			<u>Data Rights</u>			
	<u>Exist- ence</u>	<u>Manage- ment</u>	<u>Opera- tional</u>	<u>Read</u>	<u>Add</u>	<u>Update</u>	<u>Delete</u>
LIBRARY			X	X	X	X	X
PROGRAM *			X	X	X	X	X
FILE **			X	X	X	X	X
USER PROFILE			X				
DEVICE							
DESCRIPTION			X	X	X	X	X
CL COMMAND			X	X	X	X	X

* This includes object (i.e., executable) programs only. Therefore, use of data rights is limited based on a user's ability to modify machine code.

** This includes both physical and logical data files. However, Data Rights are not applicable to a logical file since a logical file has no records. See page 11 for an explanation of logical files. Source code programs also are files and each line of code is a record. Therefore, when a source code object is created, if PUBAUT is designated as *NORMAL, all user profiles can read, add to, update, and delete source code.

Default Group Profile Rights. The group authority parameter (GRPAUT) of a user profile specifies the access granted to the group profile, by default, when objects are created by this user. This parameter may be specified as *ALL, *NONE, or *NORMAL. If the desired level of access is expected to vary, the parameter can be set at *NONE or *ALL and the group rights controlled by entry of grant object authority (GRTOBJAUT) or revoke object authority (RVKOBJAUT) commands as discussed on pages 38 and 39.

The designation in the GRPAUT parameter of a user profile creating an object has no relation to the PUBAUT parameter of an object created by the user. For instance, a user may create an object and designate public authority as none (PUBAUT(*NONE)), thereby allowing the public no access. However, if the GRPAUT parameter within the user's own profile is designated as (*ALL), each member of the user's group is allowed all accesses to the object created. The level of risk associated with granting a group (or the public) *ALL or *NORMAL authority is directly related to the (1) sensitivity of the object created and (2) propriety of allowing each group member access to the object.

SECURITY FEATURES

Security Defaults. Many system conditions are determined by IBM default parameters. The parameters can be modified easily by an installation. Defaults that may cause security concerns include:

All users may read, add to, update, or delete from a database file when public authority is *NORMAL. Production database files should not have normal public authority unless access to the library that the file is in is appropriately restricted. Library security is discussed on page 45.

As installed, access to the Interactive Data Base Utilities (SEU, DFU, QRY, and SDA as described in the Utilities chapter) is not restricted. Any user has the authority to use these utilities to create application programs that could modify any data file the user can access. One method of restricting access is to place the utilities in a library that is authorized only to selected individuals.

As installed, CPF allows all users to create source programs (e.g., COBOL, BASIC, RPG III). Therefore, any user has the same abilities as those discussed for utilities. One method of restricting use of high-level programming languages is to restrict access to the various "create program" commands such as CRTRPGPGM. CRTRPGPGM allows users to create RPG programs. Revoking user access is discussed on page 39.

As discussed, all IBM-supplied user profiles are installed with the same set of passwords that are printed in the System/38 system documentation manuals. Therefore, anyone with a System/38 manual or prior experience with any System/38 may know the passwords. Installations should change all predefined user profile passwords even if the IBM-supplied user profiles are not used.

As discussed throughout the security chapters of this manual, the CPF operating system provides a variety of security measures for controlling access to objects. The alternatives should be reviewed and evaluated for each installation and the appropriate controls implemented.

Object Authority

This section discusses these aspects of object authority:

- Listing Object Authority
- Granting Object Authority
- Revoking Object Authority
- User Profile Authority

SECURITY FEATURES

Listing Object Authority. A list of authorized user profiles, and the specific rights each user has to a specified object, can be obtained by using the "display object authority" (DSPOBJAUT) CL command. Most CL commands are provided by IBM as PUBAUT(*NORMAL), noted in the chart on page 35, and allow all user profiles Object Operational Rights to those commands. If a user profile applies the DSPOBJAUT CL command to an object, the object authority display will be one of two screens:

If the requestor has Object Management Rights for the object--a complete list of authorized user profiles and their specific individual rights is provided, including the rights of the public.

If the requestor does not have Object Management Rights for the object--the list will include only the rights of the requestor, the requestor's group profile (if any), and the public.

This is an example of an OBJECT AUTHORITY DISPLAY:

```
3/17/85 15:39:20  OBJECT AUTHORITY DISPLAY
Object name: ORD040C      Library:      DSTPRODLB
Owner name:  QPGMR        Object type: *FILE

  USER NAME      |  OBJECT RIGHTS      |  DATA RIGHTS      |
  | OPER MGT  EXIST | | READ ADD  UPD  DLT |
  |-----|-----| |-----|-----|
  *PUBLIC         | | X          | | X      X      X    |
  *ADOPTED        | | X          | | X          |
  QPGMR           | | X  X      | | X  X  X  X      X  |
  KSMITH          | | X          | |          |
  BJONES          | | X  X      | |          X  X      X  |
```

The owner and those user profiles with Object Management Rights can modify object authorizations by pressing the CF3 key while viewing the object authority display. This will take the requestor to a change object authority screen. Changes are performed by on-line screen editing. For instance, an "X" indicates BJONES has "add" access to ORD040C, the owner (QPGMR) could type a blank in place of the "X". Then the enter key is pressed and access is revoked to BJONES for add rights to that object. This capability can be restricted if Object Management Rights are revoked from all users including the owner.

SECURITY FEATURES

Space also is provided for additional users to be added to the screen. The desired authorizations are designated by keying an "X" in the appropriate column.

Granting and revoking object authority rights also may be performed by using the grant object authority and revoke object authority CL commands. These commands are discussed in the next sections.

Granting Object Authority. The "grant object authority" (GRTOBJAUT) CL command is used to grant access, as needed, to objects when the designated public authority parameter does not provide the desired access. For example, if the PUBAUT parameter of an file is *NONE (or *NORMAL) and User-B needs Object Management Rights, GRTOBJAUT is used to grant User-B Object Management Rights.

Included in the GRTOBJAUT CL command is an authority parameter (AUT). This parameter may be designated as *NORMAL, *ALL, any of the Object Right and/or Data Right categories, or a combination thereof.

An example of a GRTOBJAUT CL command is shown. In this example the object is a payroll data file, PAY.FILE1. When the PAY.FILE1 object was created, PUBAUT was defined as (*NONE). The first example command gives the public read data rights. The second command gives JLONG *NORMAL authorizations, as defined in the chart on page 35 for FILES.

```
GRTOBJAUT OBJ(PAY.FILE1) OBJTYPE(*FILE) USER(*PUBLIC) AUT(*READ)
GRTOBJAUT OBJ(PAY.FILE1) OBJTYPE(*FILE) USER(JLONG)    AUT(*NORMAL)
```

This is an example of PAY.FILE1 authorizations following the entry of the CL commands.

	<u>Object Rights</u>			<u>Data Rights</u>			
	<u>Exist-</u> <u>ence</u>	<u>Manage-</u> <u>ment</u>	<u>Opera-</u> <u>tional</u>	<u>Read</u>	<u>Add</u>	<u>Update</u>	<u>Delete</u>
OWNER	X	X	X	X	X	X	X
PUBLIC				X			
JLONG			X	X	X	X	X

SECURITY FEATURES

When entering the CL command GRTOBJAUT, multiple users or public may be included in the USER parameter. Also, more than one object may be designated in the OBJ parameter or the generic indicator "*" may be used to include all common names within a group (e.g., OBJ(PAY.*)) would grant the designated authority to all file objects with the prefix "PAY").

As noted, most CL commands are supplied by IBM with PUBAUT(*NORMAL), thereby allowing all user profiles Object Operational Rights to the command. Therefore, any user with Object Management Rights to an object can apply the GRTOBJAUT command to that object. However, the authority granted may not exceed the authority of the user profile granting authority. For example, if the user profile applying the GRTOBJAUT CL command does not have Object Existence Rights, it can not grant Object Existence Rights.

By default, when an object is created, the owner and the security officer have all object rights and all data rights to that object. However, if PUBAUT is not specified as *ALL, Object Management Rights and Object Existence Rights must be granted to any other user needing these rights. Also, when a user creates an object and specifies PUBAUT(*NONE), operational and update rights must be granted using the GRTOBJAUT CL command unless the owner and security officer are the only users expected to use (e.g., execute) and update the object.

If a group profile has Object Management Rights over a specified object, each group profile member may perform all authority granting functions allowed to the group profile.

Revoking Object Authority. The "revoke object authority" (RVKOBJAUT) CL command is used to revoke authority to an object from a user profile.

Included in the RVKOBJAUT CL command is an authority parameter (AUT). This parameter may be designated as *NORMAL, *ALL, any of the Object Right and/or Data Right categories, or a combination thereof.

When entering the CL command RVKOBJAUT, multiple users or the public may be included in the USER parameter. Also, more than one object may be designated in the OBJ parameter or the generic indicator "*" may be used to include all common names within a group.

SECURITY FEATURES

As noted, most CL commands are supplied by IBM with PUBAUT(*NORMAL), thereby allowing all user profiles Object Operational Rights to the command. Therefore, any user with Object Management Rights to an object can apply the RVKOBJAUT command to that object. However, the authority revoked may not exceed the authority of the user profile revoking authority. For example, if the user profile applying the RVKOBJAUT CL command does not have Object Existence Rights, it can not revoke Object Existence Rights.

Examples of how the revoke CL command can be used are:

When a programmer transfers ownership of a program to the change control coordinator for review and transfer to the production library, the programmer retains all ownership rights. The revoke command can cancel the programmer's access rights.

When the accounts receivable supervisor transfers to the payroll department and her previous accounts receivable functions included cash correction (UPDATE), the revoke command can revoke her access to the cash data file.

Public authority is designated as *NORMAL for an inventory data file thereby allowing data read, add, update, and delete rights to all users. Management determines that the public should only be allowed to read and add data. The revoke command can revoke, update, and delete data rights from the public. (The GRTOBJAUT CL command can then be used to grant authority to the user profiles that management designates as authorized to update and delete.)

If a group profile has Object Management Rights over a specified object, each member user profile may perform all authority revoking functions allowed to the group profile.

User Profile Authority. A user profile has these rights to itself:

- Object Management Rights
- Object Operational Rights
- Data Read Rights
- Data Add Rights
- Data Delete Rights

SECURITY FEATURES

Since the user profile has Object Management Rights, the user profile can grant any of these rights to other user profiles if desired.

Even though the "rights" imply that a user profile can perform sensitive functions (e.g., add, delete) to itself, a user cannot change or delete his profile, nor can the user display the profile password. These capabilities are denied to the user profile because, as noted in the security officer discussion on page 31, the CL commands needed to perform these functions are authorized only to the security officer. The five CL commands only available to the security officer are listed on page 32.

A user profile may be displayed by any user with read rights to the user profile and operational rights to the "display user profile" (DSPUSRPRF) CL command. Using this command and either the TYPE(*BASIC) or TYPE(*ALL) parameter of the command, the requestor receives a list of all parameters (see Appendix B for a list of parameters for IBM-supplied user profiles); a list of the commands, devices, and other objects explicitly authorized to the user profile; a list of the objects owned by the user profile; and a list of the members of the user's group profile (if applicable). The display does not show the user profile password, however, it will show the date the password was last changed.

An auditor can use the DSPUSRPRF command to review the propriety of parameters and authorizations allowed to a given user, sample of users, or all users, as required. Also, a review of the timeliness of password changes can be performed.

Additional Security Features

This chapter expands on the System/38 security features discussed in the previous chapter. Each section includes either an additional capability or a complexity within a previously discussed feature.

Initial Programs

Within the user profile parameters, an Initial Program can be assigned. If an Initial Program is assigned (i.e., the parameter is not specified as *NONE), when the user signs-on to the System/38 the defined program will automatically be called. Usually the Initial Program is a menu; however, it does not have to be. The Initial Program could display a series of messages at sign-on, a complex password routine, a library assignment, or a control program consisting of a series of programs possibly ending with a mandatory menu. Menus can be used to restrict a user's processing options.

This control feature is effective only when the Initial Program provides proper limitations. For instance, if a menu called as an Initial Program allows the user to enter CL commands or has menu options that will allow access to powerful data modifying utilities, such as DFU, the effectiveness of this control feature is eliminated. Menus used as Initial Programs should be reviewed by management for options not needed by a user and options that allow a user to access programs, files, commands, displays, and libraries unrelated to the user's job responsibilities.

An example of a potential weakness in the controls provided by a menu is in the IBM-supplied Program Call Menu (see Appendix C). This menu allows the user to specify any program for use and is specified as the Initial Program in the IBM-supplied Workstation User Profile (see Appendix B). Unless other security measures are in place (e.g., public authority is designated as *NONE on all production programs), use of this user profile and menu may allow unauthorized use of sensitive programs.

System Request Key. The System Request Menu includes options that allow a user to transfer to a secondary job (e.g., exit the accounts receivable program and enter the payroll program) and suspend or cancel a job. Use of this menu is available to all users, from any menu, by pressing the System Request Key.

ADDITIONAL SECURITY FEATURES

SYSTEM REQUEST MENU	System: SYSTEM01
Select one of the following:	
1. Transfer to secondary interactive job	
2. Cancel previous request	
3. Display current job	
4. Display messages (msgq)	
5. Send message 'msg' tomsgq	
10. System request menu at source system, if BGNPASTHR active	
11. Transfer to source/target system, if BGNPASTHR active	
90. Signoff (*NOLIST *LIST)	
Option: _____	CF4 - Prompt (4, 5, and 90)
Parameters: _____	

Even if an Initial Program menu is appropriately restrictive, the System Request Key can be pressed to display the System Request Menu. One method of controlling access to the System Request Menu is to revoke public access to the menu by entering this command:

```
RVKOBJAUT OBJ(QMNSYSRQ) OBJTYPE(*FILE) USER(*PUBLIC)
```

Specific authority to the menu must then be granted to those authorized to use the System Request Menu.

Another method of controlling access is to revoke public access to the commands on the menu that allow users to perform sensitive functions.

An example is the command that allows a user to transfer to a secondary program (TRSECJOB). Revoking public use of this command may have an adverse effect on the activities of users with legitimate needs to transfer to a secondary job. In those situations, specific authority can be granted to those individuals.

An auditor can determine if access to the System Request Menu (QMNSYSRQ) is properly restricted by using the "display object authority" (DSPOBJAUT) CL command (discussed on page 37), to display users with access to QMNSYSRQ.

Screen Design Aid (SDA) Menus. If an Initial Program menu was created using SDA, and the user presses the CF1 key, the CL command entry display is called for use by the user. The command entry display is discussed on page seven. It can be used by a user to execute any CL command that has public authority *ALL or *NORMAL. Unless modified after system installation, most CL commands have PUBAUT(*NORMAL). Access to sensitive commands may be restricted using several methods, one method is to restrict access to the displays that allow commands to be entered.

ADDITIONAL SECURITY FEATURES

Access to the command entry display in this situation may be prevented by several methods. One method is for the installation to modify the SDABEGIN control language program. Modify "IF COND(&IN01) THEN(GOTO SDAEND)" to "IF COND(&IN01) THEN(GOTO SDABEGIN)". This causes the program to branch back to the SDA menu rather than the command entry display when CF1 ("&IN01") is pressed.

Program Adopt Function

When a user executes a program, that user must have:

- Rights to the program.

- Rights to the programs called by the original program.

- Rights to each file that each program accesses.

This path is called the invocation stack. Granting each individual authorized to use a program the appropriate access to all objects needed to execute the program to completion can be a significant task. Therefore, an "adoptive" function is provided with the system.

The object description for each program has a USRPRF parameter that designates the user profile under which the program will execute. If the USRPRF parameter of a program is defined as *USER, and a given user executes the program, it may only access objects authorized to that user.

However, assume the owner of the original program also owns all objects needed to execute the program to completion. If the USRPRF parameter of the program is designated as *OWNER, the program runs under the owner's profile regardless of who executes the program. An "adoptive user" must then be granted authorization to execute only the original program. When the adoptive user executes the program the user appears to the system to be the owner, and can therefore access all needed objects. The adoptive user's access to these objects is only valid when executing this program.

Among the control concerns relating to the use of adoptive programs are:

- If the program creates an object, such as a data file, the owner of the object is the adoptive user actually executing the program, not the owner of the program. Therefore, future

ADDITIONAL SECURITY FEATURES

users of the program will not have access to the new data file including the original program owner. With some planning the problem can be avoided by having the owner create shells (i.e., object descriptions without records) for the objects before the user executes the program.

If a program allows the adoptive user to select menu options not needed to complete processing, this may provide the adoptive user with access to sensitive data or programs outside the scope of the intended procedures. Since adoptive authorization is similar to a path, it is important that the branches off the main path are limited. An example would be a program that allowed a user to access DFU to create a program. Using DFU the user can access all objects the adopted owner (or user) can access.

Using the "display programs that adopt" CL command (DSPPGMADP) an auditor can obtain a listing of programs by owner that have USRPRF(*OWNER) specified. The listing can then be reviewed for potential security risks after consideration of program capabilities and the owner's authorizations.

Libraries

A library within the System/38 is an object. All other types of objects are included in a library. Any combination of other objects can be placed in a particular library. For instance, all production versions of programs and data files could be placed in one library and test versions in another library. Access to the production library could then be granted only to authorized personnel.

A user must have rights to the library containing an object before the object can be accessed. For example, if a database within a library is defined as PUBAUT(*ALL) but the library is defined as PUBAUT(*NONE), only those users with explicit rights to the library can access the database.

Access to CL commands, compilers, and utilities can be restricted using library security by placing these objects in libraries with PUBAUT(*NONE) and explicitly granting access to the objects to authorized users.

ADDITIONAL SECURITY FEATURES

Use of libraries can simplify security for an installation. Various security methodologies are:

Restrict production library access to a designated group of users and allow public authority (*ALL) or (*NORMAL) to all objects within the library;

Restrict production library access to a designated group of users and restrict access to objects in the library to certain users within that group; or

Combination of the two methods.

A list of all libraries on the system and a list of all objects in a library by type (e.g., command, file, program) may be obtained using the DSPLIB CL command. As installed the System/38 system library (QSYS) includes over 2500 objects. The QSYS library list includes each CL command (each a separate object) with a short description.

Logical Files

Access to sensitive data fields can be restricted by use of logical files. Logical files that exclude the sensitive fields contained in physical files can be created. User profiles can then be allowed to access only the logical files. Logical files are discussed on page 11.

For example, a physical file named PAY.FILE1 may include fields such as employee pay rates, hours worked, and employee number. Instead of using the physical file, PAY.FILE1, a program updating employees hours could use a logical file called PAY.FILE2. The logical file would include only the employee number and hours worked fields of the physical file. This would restrict user access to the pay rate field when using the program that updates hours worked.

Save and Restore

Any object saved or restored on one System/38 can be restored (copied) to a different System/38. The object authorities restored to the second machine will vary depending on who performs the restore procedure and whether or not the user profiles that are authorized to the restored object exist on the second system. In some situations all authorities will be restored, in others no authorities will be restored. If needed, details of those

ADDITIONAL SECURITY FEATURES

conventions may be obtained from the "Save/Restore Operations" chapter of IBM's CPF programmer's guide (see Bibliography).

An installation's save/restore procedures should include:

Procedures that ensure all objects restored from other systems do not undermine the security goals of the installation. An example of a potential risk is restoring a program, created on a vendor System/38, with public authority specified as PUBAUT(*ALL) that contains code to modify sensitive data files.

Use of the save/restore functions should be restricted. User profiles with Special Authority Rights can save/restore (copy) objects to diskette or tape--which are easily transportable media. An example of a potential risk is sensitive data saved/restored then viewed on another system by employees or possibly sold to a competitor. Physical access to all off-line media should be restricted for the same reason.

The CL restore commands (e.g., restore library (RSTLIB)) should be restricted to authorized individuals. In particular, the restore user profile CL command (RSTUSRPRF) used to restore the user profile objects should be authorized to the security officer exclusively to ensure confidentiality of user profile information.

Save/restore procedures and authorities may be of particular interest if a client maintains more than one System/38 and transfers objects (e.g., data, programs) between systems.

Console/Workstation Access

Console. The System/38 console is equipped with external rotary switches that can be set to allow anyone to display or alter main storage. These switches are on the exterior of the console and look like dials. The switches can not be locked; therefore, to control use of the switches, access to the console itself must be controlled.

To effectively use the switches, the user must be technically proficient. Although this may limit the risk that data will be altered intentionally, access to the switches does provide an opportunity for experimentation and sabotage.

The security officer profile (QSECOFR) and the two IBM service profiles (QCE and QPSR) are allowed use of the console by default. This authority cannot be revoked from the security officer. User profiles should be authorized to use the console only as needed (i.e., most user profiles should not be authorized to sign-on to the console).

ADDITIONAL SECURITY FEATURES

The QCE and QPSR profiles include the ability to alter main storage; therefore, access to these profiles should be restricted. As stated, the passwords supplied at installation should be changed as soon as possible. However, care should be taken to ensure the new passwords are available to the service representatives when needed. Changing the vendor passwords after the completion of each service visit may restrict vendor access to the system without authorization.

Most system console functions can be performed from any workstation. A discussion of system operator functions is included on page 16.

Workstations. Some IBM terminals provide external key locks that can, if properly used, control access to terminals when unattended. If terminals are equipped with key locks, installations should require that terminals be locked when not in use.

Each workstation on the system must have a device description object. The device description specifies terminal parameters (e.g., type, location, owner, authorizations) to the system. A user must either be explicitly authorized to use a workstation or the workstation device description must have public authorization of *ALL or *NORMAL.

Each device description has an owner; and, object rights can be defined to specify who can access the workstation.

The security officer user profile is not authorized to use to all workstations by default. The security officer is authorized to use:

The system console.

Workstations "owned" by the security officer.

Any workstation the security officer has been granted access to by a user with Object Management Rights over that device (workstation).

Any workstation the security officer has granted itself access to (in this situation, the security officer uses another workstation authorized to the security officer to grant itself access to this workstation).

If a workstation device description is created (by someone other than the security officer) and public authority is designated as

ADDITIONAL SECURITY FEATURES

*NORMAL, the security officer and the two IBM service profiles are not granted access by default. Specific authority must be granted to these profiles to allow use of the workstations. This discourages experimentation with these user profiles (e.g., an unauthorized user attempting to guess the security officer's password). An auditor can use the DSPUSRPRF CL command to display the security officer's profile to determine the workstations authorized for security officer use.

Passwords

When a user attempts to access the system, a password is required if the USER parameter of the job description is designated as *RQD. A job description is an object that defines the attributes of a job. For instance, for the interactive subsystem QINTER (discussed on page 9) the related job description object, also QINTER, must have *RQD specified in the USER parameter to require a password of all user profiles using the subsystem. All interactive sessions are performed in a subsystem.

If USER is not defined as *RQD, a user profile name is entered for this parameter. This allows anyone to merely press the enter key to gain access to the system. All users using the subsystem are represented to the system as the user profile named in the USER parameter of the job description.

For example, if the job description for the subsystem QINTER defined the USER parameter as QPGMR (USER(QPRMR)) (see user profile for IBM predefined "Programmer User Profile (QPRMR)" in Appendix B) anyone could hit the enter key and have access to all objects accessible to the QPGMR profile.

This approach may be appropriate for a nonsensitive subsystem used for inquiry only. For subsystems used to process significant accounting applications and/or maintain sensitive objects, all USER parameters of related job descriptions should be defined as *RQD. The remainder of this section relates only to subsystems where passwords are required.

A user profile object name and/or a password are required to sign-on to the system. Conventions related to sign-on vary based on the value specified in the operating system value "QSIGNLVL." These conventions are:

ADDITIONAL SECURITY FEATURES

QSIGNLVL

<u>Value</u>	<u>Options</u>
--------------	----------------

- | | |
|---|--|
| 0 | Password only is required to sign-on to the system. (Two user profiles cannot be assigned the same password, see narrative.) The password may be viewed by the security officer user profile and the IBM customer engineer using the "display user password" (DSPUSRPWD) CL command. |
| 1 | User profile object name and password are required to sign-on to the system. Passwords are encrypted by the system. Passwords cannot be viewed by any user including the security officer and IBM user profiles. |

If "0" is specified, users should not be allowed to assign their own password because the system will not allow duplicate passwords. If a user were to attempt to change their password and inadvertently selected a password already in use, the system would inform them of that fact. The user would then have knowledge of another user's password.

If "1" is specified, users can select their own passwords without compromising the security of the installation. (The installation must provide the users with a program to do so because it is not an available system option.) Passwords can be duplicated without the users knowledge because the system uses the combination of the user profile object name and the password to identify the user.

The system value "QMAXSIGN" specifies the number of unauthorized password attempts that may occur before a terminal is deactivated. Once disconnected the system operator must reactivate the terminal. This should be done only after an explanation of why the deactivation occurred. As installed, the system default allows 32,767 invalid access attempts.

Operating system values (e.g., QSIGNLVL, QMAXSIGN) may be changed using the "change system value" (CHGSYSVAL) command; however, the change will not be in effect until after the next system initialization.

If an unsuccessful attempt to enter a password is made, the system history log receives an informational message (CPF2234). The message indicates from which workstation the attempt was made. Unsuccessful access attempts can be reviewed using the "Display Log" CL command:

```
DSPLOG LOG(QHST) MSGID(CPF2234)
```

ADDITIONAL SECURITY FEATURES

See page 17 for a history log excerpt.

The system does not include an option to require passwords to be changed periodically. However, the date of last change is included in each user profile.

Split Password. The Security Chapter of the IBM System/38 Control Program Facility Programmer's Guide (SC21-7730 CPF Release 7, Chapter 21, pages 21-56 through 21-65) contains coding for a "split password" program. The term "split password" refers to a password composed of a two-digit code that remains constant and an eight-digit password. Basically, if the installation implements this program or any other similar program, password controls can be improved. The program provides these options not available on the System/38 as installed:

Passwords must be changed periodically. The minimum time frame is defined in the program; if the minimum time passes and the password has not been changed, the user is forced to change the password.

A user may select his or her own password; however, that selection may be limited by a history file of previously used passwords that cannot be reused by the user. This also can be used to restrict the user from using his or her name or other common objects as a password.

Passwords selected must contain a minimum of "n" consecutive nonblank characters ("n" to be designated by the installation).

Workstation Inactivity

The System/38 does not automatically deactivate terminals after a stated time frame. However, a job that will poll the terminals for activity can be created and then executed periodically by using the "delay job" (DLYJOB) CL command. If no activity occurs in the defined time frame, the device can be varied off the system (i.e., deactivated) using the VRYDEV CL command.

Subsystems and communication lines also can be varied off the system using the "terminate subsystem" (TRMSBS) CL command and the "vary line" (VRYLIN) CL command, respectively. These commands also may be used to control access to the system after regular business hours.

Procedural and Administrative Controls

An effective combination of the automated control procedures offered by the System/38 and manual control procedures is required to control (1) program development, acquisition, and maintenance and (2) access to data files. This section identifies procedural and administrative controls that should be considered at System/38 installations. The controls are not meant to be a comprehensive discussion, nor is it necessary for an installation to have all the controls for the system of internal controls to be effective. The auditor's overall evaluation of the system is a matter of judgement determined only after the risks at a particular installation are assessed.

Organization and Administration

The organization of a data processing department should be designed to provide segregation between operations, systems and applications programming, and data control. Often in System/38 installations there are a limited number of data processing personnel. Therefore, control concerns regarding the segregation of duties within data processing may result. Examples of mitigating controls that may address or monitor a lack of segregation of duties are:

Programmer access to production objects is limited to read only by using available CPF operating system security features.

Access to source production programs and compilers is restricted using available CPF operating system security features. Each access to these objects is permitted only with management's authorization.

The history log is reviewed by management for unauthorized use of system libraries, utilities, and compilers. Unusual activity is investigated.

Operations personnel are restricted to an "Initial Program" that restricts processing options by design.

Object descriptions, which include date of last change, are periodically compared to approved program change forms.

Physical access to the computer and the system console is limited to authorized personnel.

PROCEDURAL AND ADMINISTRATIVE CONTROLS

If no in-house programming is performed, use of purchased software or third-party contract programmers may provide a segregation of duties between programming and the installation's operations and user departments.

Examples of mitigating controls that may address or monitor a lack of segregation of duties between user departments and data processing are:

Users are assigned "Initial Programs" (e.g., a mandatory menu) that restricts options available to each user.

Management restricts programmers from accessing production data files by using available CPF operating system security features.

An effective system of reconciling inputs and outputs, such as the use of batch controls and the review of transaction listings, is in place. Personnel independent of authorizing and entering transactions are responsible for reconciliation and review procedures.

Program Development, Acquisition, and Maintenance

An installation should have standards for procedures over program development, acquisition, and maintenance. An adequate system of establishing standards and reviewing adherence to standards provides controls that assure production programs are authorized, tested, and approved prior to being placed in production. Controls that can be achieved by establishing and adhering to standards are:

Requests for program development, acquisition, and maintenance are standardized and approved by data processing management and by users, as appropriate.

All program development, acquisition, and maintenance is adequately tested and reviewed by data processing management and by users, as appropriate, before implementation into production.

All programs and changes to programs are properly documented and documentation is reviewed for propriety.

PROCEDURAL AND ADMINISTRATIVE CONTROLS

The SEU date stamp on each source code line is reviewed to determine that changes made are in accordance with changes authorized.

Separate libraries are maintained for (a) programmers to develop or modify programs, (b) testing of new or changed programs, and (c) production programs.

Source Entry Utility (SEU) is not used to create/update production source code (all source code is created/updated in a test environment).

Programmers are restricted from adding new or changed programs to the production library. Only specified operations or data processing management have Object Management Rights to production objects (the right to move, rename, grant authority to, revoke authority from, and modify the attributes of an object).

Procedures related to the acquisition and subsequent modification of purchased software also should be subject to controls that require user and data processing approval and appropriate testing before the programs are placed in production. Procedures should be performed whether modifications are done in-house or by outside vendors.

Access to Data Files

The installation should have procedures that ensure access to data files is restricted to authorized users and programs. The System/38 provides a security system that can be used to restrict access; however, the effectiveness of the system depends on how it is implemented and used.

Examples of installation procedures and controls are:

Object access rights granted to users are documented and approved by their supervisor.

Each individual is assigned a unique profile and password.

Passwords are confidential and difficult to decode.

Passwords, including those of the security officer and outside vendors, are changed periodically.

PROCEDURAL AND ADMINISTRATIVE CONTROLS

Passwords are encrypted (the installation has specified "1" in the operating system value (QSIGNLVL)).

Workstations are deactivated after a reasonable number of password violations. (The number of attempts allowed is designated in the system value (QMAXSIGN)).

Access violations are investigated by appropriate management personnel.

Access to hard copies of password listings, if any, is restricted (only available if QSIGNLVL is specified as "0").

The security officer profile and the security officer profile as a group profile are only assigned to a limited number of management personnel who have security responsibilities.

Security functions may be performed only from a limited number of terminals.

Users are assigned an "Initial Program" that limits access to that needed to perform their work.

Access to the system is controlled after business hours through the use of VRYDEV (for devices), TRMSBS (for subsystems), and VRYLIN (for communication lines) commands.

Access to removable data files (e.g., diskettes, tapes) is restricted to authorized personnel.

The system is programmed to cancel an interactive job (i.e., terminal session) if there is a specified period of inactivity.

System Save Rights and Job Control Rights are limited to authorized personnel.

Authorization to use restore commands is limited to appropriate personnel.

Access to the System Request Menu is limited to appropriate personnel.

Use of data altering utilities (e.g., DFU) is restricted to authorized personnel and their usage is closely monitored.

Jobs are executed during scheduled time frames and deviations from scheduled processing is monitored and reviewed.

Control Concerns and Audit Procedures

This section identifies control concerns and related tests that may be performed at a System/38 installation. The concerns and tests are not meant to be a comprehensive discussion, nor is performing all the tests necessary in every situation. The selection of procedures to be performed, if any, is a matter of judgement determined only after the risks at a particular client installation are assessed.

Control Concern: Access to production objects should be restricted.

Audit Procedures:

Using the "display system value" (DSPSYSVAL) CL command ascertain the system value for "QSIGNLVL." If QSIGNLVL is specified as "0", determine that password lists, if any, are safeguarded. Reference page 50.

Review password conventions for propriety. Reference page 49.

Identify the objects (e.g., libraries, programs, data files) related to significant accounting applications and determine that passwords are required to access significant accounting applications by noting that USER(*RQD) is specified in the job description of the subsystem processing the applications (in many installations, QINTER will be the subsystem). Reference page 49.

Determine that production objects are maintained in a separate library(s). Request the "Object Authority Display" for each production library; evaluate the propriety of the profiles allowed access and the level of access allowed (e.g., read, delete, add). Determine that access levels provide appropriate segregation of duties between programmers, operators, and users (e.g., programmers are allowed read only access). Determine that the PUBAUT parameter of each production library is *NONE. These procedures should be performed at the object level if access to libraries is not restricted. Reference pages 34, 37, and 45.

Identify data altering utilities (e.g., DFU, SEU, DIF, SDA) and compilers, and determine that access is restricted and monitored. Request the "Object Authority Display" for each; evaluate the propriety of profiles allowed access and the level of access allowed. Determine that users cannot use utilities to modify production programs or data. Identify

CONTROL CONCERNS AND AUDIT PROCEDURES

individuals responsible for monitoring access and review documentation of related review procedures performed by the installation and evaluate propriety of procedures performed. Reference pages 20 and 37.

With client assistance, attempt to log-on to sensitive objects using unauthorized user profiles. Review the history log (QHST) for logging of the attempt. Review documentation of related review procedures performed by the installation and evaluate propriety of procedures performed. Reference page 17.

With client assistance, attempt to log-on to the system using invalid passwords. Attempts should exceed the (QMAXSIGN) system value to test deactivation of the terminal due to violation attempts. Review the history log (QHST) for logging of the unauthorized access attempts. Review documentation of related review procedures performed by the installation and evaluate propriety of procedures performed. Reference pages 17 and 50.

Determine that each user has a unique user profile. If a group of users share a profile (i.e., share a password) determine that the user profile is allowed read accesses only including authorities allowed to the public. For a sample of users, review documentation supporting supervisory approval for the user to access the system. Reference page 26.

Examine documentation of the regular review of the history log (QHST). Determine that review is designed to provide for both detection and follow-up of unauthorized access attempts, unauthorized use of system utilities and compilers, and unscheduled processing. Reference page 17.

Control Concern: Certain Control Language commands allow users to perform sensitive functions. As installed, access to most commands is not restricted.

Audit Procedure:

Inquire of client personnel how access to sensitive CL commands is restricted (e.g., all sensitive CL commands are placed in a library that is restricted to authorized personnel, the PUBAUT parameter (within the command's object description) has been specified as *NONE). Review and test implementation of client controls for propriety. Reference page 6.

CONTROL CONCERNS AND AUDIT PROCEDURES

Control Concern: Use of the group profile user profile parameter allows all group members access to all objects available to the group profile. All group accesses should be appropriate for all group members.

Audit Procedure:

Using the "display authorized users" (DSPAUTUSR) CL command determine which user profiles function as group profiles. For each group profile (or a sample) obtain a list of all group members and objects authorized to the group profile (use the "display user profile" (DSPUSRPRF) CL command). Review the group members in relation to the objects authorized to the group profile for propriety. Reference pages 29, 32, and 41.

Control Concern: Use of the Program Adopt Function may inadvertently allow unauthorized users access to objects.

Audit Procedure:

Using the "display programs that adopt" (DSPPGMADP) CL command, obtain a list of programs that adopt by owner. For significant accounting applications, review the objects authorized to the owner's user profile in conjunction with the program capabilities to determine if access is appropriate for all users using the adopted program. Reference pages 44 and 45.

Control Concern: User profiles have all accesses to objects they create unless accesses are revoked by the security officer or a subsequent owner (if ownership was transferred). In certain situations, authorizations allowed the current or previous owners should be revoked (e.g., a programmer creates a program and is therefore the owner, when the program is reviewed and transferred to the production library, ownership should be transferred and all accesses except read should be revoked).

Audit Procedure:

Determine what procedures are performed by an installation to assure that current or prior "ownership" of an object does not compromise installation security policies. Determine the effectiveness of client procedures by reviewing user profiles (includes objects owned) and/or object authority displays (includes object owner). Reference page 34.

CONTROL CONCERNS AND AUDIT PROCEDURES

Control Concern: Certain system-defined profiles provide unlimited access to virtually all aspects of the System/38.

Audit Procedures:

Review all responsibilities of the individual assigned the (QSECOFR) profile for appropriate segregation of duties. Using the display user profile command (DSPUSRPRF) obtain a listing of security officer group profile members; evaluate the propriety of each recognizing that all have access to all aspects of the system. Reference pages 31 and 41.

Determine that the installation has changed the IBM-assigned password (reference page 28) for these user profiles:

Security Officer	(QSECOFR)
Programmer	(QPGMR)
Workstation User	(QUSER)
System Operator	(QSYSOPR)
Programming Service Representative	(QPSR)
Customer Engineer	(QCE)

Using the display user profile command (DSPUSRPRF) evaluate the propriety of each workstation owned or authorized for use by the security officer profile (QSECOFR). Reference pages 31, 41, and 48.

Control Concern: The contents of a file can be changed without leaving an audit trail using the Data File Utility (DFU) and Display Information Facility (DIF).

Audit Procedures:

Obtain the "Object Authority Display" for the DFU and DIF program objects. (If access is controlled at a library level, obtain the "Object Authority Display" for the library that contains the DFU and DIF program objects.) Determine that the PUBAUT parameter is *NONE and that individual authorities listed are appropriate. Review documentation of review procedures performed by the installation that monitors usage of these objects and evaluate propriety of procedures performed. Reference pages 22, 24, 34, and 37.

Control Concern: If Initial Programs are not assigned to user profiles or are not properly restrictive, users may be able to access sensitive system commands or other objects not otherwise restricted.

CONTROL CONCERNS AND AUDIT PROCEDURES

Audit Procedures:

With client assistance and using the "display user profile" (DSPUSRPRF) CL command, determine that users profiles have been assigned an Initial Program. (Based on installation size, this procedure may be performed on a sample of user profiles.) Evaluate the propriety of the Initial Program assigned based on the individual's job description. Review the Initial Program assigned (usually a menu) to determine that options that allow the user to access programs or data files do not conflict with the installations segregation of duties conventions. Reference pages 41 and 42.

Control Concern: Public authority for database files and source code files when created defaults to PUBAUT(*NORMAL); this allows all users to read, add to, delete from, and update these files.

Audit Procedures:

Determine that the production database files and production source code files are maintained in a library with appropriately restricted access. Or, using the "Object Authority Display," determine that the PUBAUT parameter for each significant individual production database file and production source code file is *NONE and individual accesses allowed are appropriate. Reference pages 34, 37, and 45.

Control Concern: Rotary switches located on the exterior of the system console can be set to allow anyone to display or alter main storage.

Audit Procedures:

Review installation procedures that limit access to the system console and thereby limit the opportunity for unauthorized display or alteration of main storage in addition to deterring sabotage. Reference page 47.

Control Concern: Objects maintained off-line (e.g., tapes, diskettes) may be restored to another System/38 where security can be circumvented.

CONTROL CONCERNS AND AUDIT PROCEDURES

Audit Procedures:

Review installation procedures that limit access to off-line files and thereby limit the opportunity for unauthorized display or alteration of programs or data. Review manual logs maintained to monitor off-line files. Compare information on the log to the labels on the actual tapes and/or diskettes. Reference page 46.

Determine that the special authority parameter SPCAUT within the user profiles is designated as *SAVSYS and/or *JOBCTL only for those individuals authorized by management. Determine that restore commands are appropriately restricted to authorized individuals. Reference pages 27 and 46.

Appendix A—Using Query as a Computerized Audit Tool

Query and Data File Utility (DFU) are components of System/38 "Interactive Data Base Utilities." This utility package is usually purchased in conjunction with the acquisition of a System/38. Although DFU can be used to perform many of the audit procedures included in this appendix, Query should be used instead because it is easier to use and does not compromise the integrity of data files (i.e., DFU can be used to update files without leaving an audit trail, but Query can only be used for extraction purposes).

Query is a menu-driven program that allows users to sort, total, extract, and summarize data. No data processing experience is required to use Query. Verbal instructions from a Query user or review of the IBM System/38 Query Utility Reference Manual and User's Guide (see Bibliography) provides adequate instruction on successful use of Query.

To use Query as an audit tool, request the security officer to assign a user profile for audit use. Access to Query and read access to any data files to be tested should be granted. If the audit procedures cover a time frame or point of time far removed from the date the procedures are to be performed, the installation should be contacted to ensure that copies of the data files are available when needed.

Using Query menus and prompts an auditor can:

- Write, modify, delete, move, or execute a Query program.

- Print or display a Query report.

- Give authorization to other users to use auditor-written Query programs.

- Select a sample of records.

- Chain files together for reporting use (e.g., have a Query program combine the payroll information from different databases maintained by region for use in one company-wide report).

- Sort files based on criteria set by the auditor.

- Rank, average, and total data.

- Design reports in the form desired.

- Extract data based on conditions such as equal to, greater than, less than, etc.

- Create matrix presentations of data.

Appendix B—IBM-Supplied Profiles

The System/38 is installed with several predefined profiles. These profiles and the preset parameters are included in this appendix. Installation use of the supplied profiles should be challenged because the design may not provide for adequate user accountability and segregation of duties—particularly in situations where multiple users share a profile (i.e., share a password) and the Initial Program assigned allows the user to access options conflicting with their job description.

The security officer (QSECOFR), IBM programming service representative (QPSR), and IBM customer engineer (QCE) profiles should not be readily available to more than one user and in all cases the IBM-assigned passwords for all profiles should be changed immediately after the system is installed.

These parameters are described in the User Profiles section of the Security Features chapter of this manual. The IBM-supplied profiles and related parameters are:

Security Officer User Profile (QSECOFR)

Password: SECOFR

Special Authority: *SAVSYS, *JOBCTRL, *ADMIN

Maximum Auxiliary Storage: *NOMAX

Highest Scheduling Priority: 1

Initial Program: *NONE (defaults to Command Entry Display)

Job Description: *NONE

Group Profile: *NONE

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *Blank

DOCPWD: *NONE (A)

MSGQ: *NONE (B)

OUTQ: *NONE (B)

Programmer User Profile (QPGMR)

Password: PGMR

Special Authority: *SAVSYS, *JOBCTRL

Maximum Auxiliary Storage: *NOMAX

Highest Scheduling Priority: 5

Initial Program: QPGMMENU (Programmer Menu)

Job Description: *NONE

Group Profile: *NONE

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *Blank

DOCPWD: *NONE (A)

MSGQ: *NONE (B)

OUTQ: *NONE (B)

APPENDIX B--IBM-SUPPLIED PROFILES

System Operator Profile (QSYSOPR)

Password: SYSOPR

Special Authority: *SAVSYS, *JOBCTRL

Maximum Auxiliary Storage: 6348 K

Highest Scheduling Priority: 3

Initial Program: QOPRMENU (System Operator Menu)

Job Description: *NONE

Group Profile: *NONE

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *Blank

DOCPWD: *NONE (A)

MSGQ: *NONE (B)

OUTQ: *NONE (B)

Workstation User Profile (QUSER)

Password: QUSER

Special Authority: *NONE

Maximum Auxiliary Storage: 3174 K

Highest Scheduling Priority: 5

Initial Program: QCALLMENU (Program Call Menu)

Job Description: *NONE

Group Profile: *NONE

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *Blank

DOCPWD: *NONE (A)

MSGQ: *NONE (B)

OUTQ: *NONE (B)

Programming Service Representative Profile (QPSR)

Password: PSR

Special Authority: *SAVSYS, *JOBCTRL

Maximum Auxiliary Storage: *NOMAX

Highest Scheduling Priority: 3

Initial Program: *NONE (defaults to Command Entry Display)

Job Description: *NONE

Group Profile: *NONE

Owner: *USRPRF

Group Authority: *NONE

Accounting Code: *Blank

DOCPWD: *NONE (A)

MSGQ: *NONE (B)

OUTQ: *NONE (B)

APPENDIX B--IBM-SUPPLIED PROFILES

Customer Engineer Profile (QCE)
Password: CE
Special Authority: *JOBCTRL
Maximum Auxiliary Storage: *NOMAX
Highest Scheduling Priority: 3
Initial Program: QSMCSMSU (Concurrent Service Monitor)
Job Description: *NONE
Group Profile: *NONE
Owner: *USRPRF
Group Authority: *NONE
Accounting Code: *SYS
DOCPWD: *NONE (A)
MSGQ: *NONE (B)
OUTQ: *NONE (B)

(A) This relates to an auxiliary software product available for the System/38 called OFFICE/38. Because this software is optional and generally has little impact on significant accounting applications, use of this feature is not discussed in this manual. Additional information about the product is available in various IBM/38 Office/38 manuals provided with the Office/38 system.

(B) These are the default messages and output queues related to this user profile.

Appendix C—Selected IBM-Supplied Menus

The System/38 is installed with several predefined menus. Use of the supplied menus should be challenged because the design may not provide for adequate segregation of duties—particularly in situations where the Initial Program assigned to a profile allows the user to access options conflicting with their job description. Each of these menus is included as an Initial Program in an IBM-supplied user profile as shown in Appendix B. Also, each allows a user to enter Control Language (CL) commands. As noted in the Security Features chapter, unless access was modified after the System/38 was installed, use of most CL commands is not restricted. Examples of the IBM-supplied programmer, operator, and program call menus are:

```

                                PROGRAMMER MENU
Select one of the following:
1. Design/execute DFU app      (app), ,(options)
2. Design/execute query app   (app), ,(options)
3. Create object              object name, type, pgm for CMD, (text)
4. Call program                program name
5. Execute command            command
6. Submit job                  (job name), (command)
7. Display submitted jobs
8. Edit source                 (srcmbr), (type), (text)
9. Design display format      (srcmbr)
80. Display menu              (menu)
90. Sign off                   (*NOLIST *LIST)
Types: BAS, BASP, BSCF, CBL, CL, CLP, CMD, CMNF, DFU, DSPF, LF, MXDF, PF, PRTF,
      QRY, RPG, RPT, TXT
Option: 5 Parm: _____ Type: _____ Parm 2: _____
Command: CHV,IB

Text: _____ Log requests: *YES
Src file: _____ Src lib: *LIBL Obj lib: _____ Jobd: QBATCH _____
CF3-Command entry CF4-Prompt (3,5 & 6 only) CF6-DSPMSG
```

```

                                SYSTEM OPERATOR MENU
Select one of the following:
1. DSPJOBQ (jobq)
2. DSPOUTQ (outq)
3. SNDMSG tomsgq,(type),msg
4. CALL program                command
5. Execute command            command
6. SBMJOB (job),(jobd),(cmd)
7. STRPRTWTR device,outq
8. DSPWTR (writer)
9. SBMDKJOB dev,label,(loc)
10. SBMOBJOB file,(member)
11. DSPSBMJOB
12. DSPACTJOB (reset)
80. DSPMNU (menu)
90. SIGNOFF (*NOLIST *LIST)
Option: 5 Parm: _____
Cmd or parm: _____

Log requests: *YES CF3-Command entry CF4-Prompt (5,6 only)
CF6-DSPMSG QSYSOPR CF7-DSPSBS CF8-DSPSYS
```

```

                                PROGRAM CALL MENU
Select one of the following:
1. Call program (identify below)
2. Display messages
3. Send message to system operator
90. Sign off work station (*NOLIST *LIST)
```

```

Option: 1 Program name: CHV,IB
Parameters or message: _____
```

Glossary

Adopt	Relates to a user that is executing a program adopting the rights held by the owner of the program. This occurs if the USRPRF parameter of the program's object description is designated as *OWNER. This allows an otherwise unauthorized user to execute the program.
Batch Processing	A technique of processing a record or series of records (a batch) together in a group. Records are generally accumulated and processed by submitting a Control Language command. Once submitted, limited user interaction is required. Contrast with interactive processing.
Commitment Control	A method of grouping a series of related transactions together for posting after all related transactions have been entered rather than at the time each transaction is entered. This aids in ensuring the information in different areas of the database are consistent at a point in time.
Communications (Software)	Software used to facilitate the transmission of data from one device to another device located in a different physical location.
Compiler	A program that translates source code (e.g., COBOL, FORTRAN) to object code (machine language).
Control Language (CL)	The commands or instructions used to run the Control Program Facility (CPF). Also called "command language."
Control Program Facility (CPF)	The operating system for the IBM System/38.

GLOSSARY

Data Description Specifications (DDS)	The description of a database file. A DDS includes items such as fields, field length, record type, and record length.
Data File Utility (DFU)	An IBM utility program that can be used to update and read data files. DFU can be used without leaving an audit trail.
Data Management	A component of the CPF operating system that provides for the creation, deletion, and manipulation of data files and device files.
Data Rights	The right or authority to add to, delete from, modify, and/or read a data file.
Default	A value automatically specified by the system unless specifically modified.
Device Description	An object that describes a peripheral device to a System/38.
Display	A terminal screen.
Display Information Utility (DIF)	An IBM utility program generally used to simplify programming. DIF can be used to inquire, add to, delete from, modify, search, and list files.
Field Reference File	A DDS, similar to a data dictionary, used to define data fields to the system. A field reference file contains no records.
Group Profile	A user profile referenced in the GRPPRF parameter of a second user profile thereby providing the second profile with all the authorizations of the first.
Initial Program	The program designated in a user's profile that automatically appears when that user signs on, usually a menu.

GLOSSARY

Interactive Data Base Utilities	An IBM utility package consisting of the Data File Utility, Source Entry Utility, Query, and Screen Design Aid.
Interactive Program	Conversational processing. A user inputs and the system responds resulting in a dialogue. An interactive job (session) begins at user sign-on and ends at sign-off.
Job Control Rights	A Special Authority that relates to the system overall rather than to specific objects. Job Control Rights specifically allow the user to release, hold, cancel, and display jobs submitted.
Journal Management	A component of the CPF operating system that can record changes made to a file for use in reconstructing or auditing transactions at a later time.
Library	An object that serves as a directory of other objects. Access to objects in a library may be controlled at the library level.
Logical File	Defines how selected fields within a physical database file or files are organized for use by a program or user. Although it appears to be a physical file to the program using it, a logical file does not actually contain data records. Logical files are often used to provide security over sensitive data fields in physical files.
Object	A general term for all named items stored on the System/38. Examples include programs, libraries, and data files.

GLOSSARY

Object Authority	The rights related to accessing and using an object.
Object Existence Rights	The authorization to delete, save, restore, and transfer ownership of an object.
Object Management Rights	The authorization to move, rename, grant authority to, revoke authority from, and modify the attributes of an object.
Object Program	(1) The compiled (machine readable) version of a program. (2) A System/38 object that is a program.
Object Rights	The rights a particular user has to access an object. Object rights include Object Management Rights, Object Existence Rights, and Operational Rights.
Operational Rights	The right to look at an object's description and to use (e.g., execute) the object.
Owner	The creator of an object, unless ownership has been transferred to another user. Owners have all Data Rights and all Object Rights to an object.
Physical File	Composed of (1) a description of how these data (records) are organized and (2) the actual records described. Contrast with logical file.
Profile	See User Profile.
Public Authority	The specified access to an object allowed to all users of the system.

GLOSSARY

Queues	A "line" of items waiting for service, such as reports waiting to print. Usually based on the priority level of the user and the order the requests are submitted.
Query (Utility)	An IBM report writing utility program used to read and extract, for display or print, information from a data file.
Restore	The reconstruction of an object on main storage performed by copying from tape or diskette.
Screen Design Aid (SDA)	An IBM utility that simplifies the on-line programming of menus and display formats.
Source Entry Utility (SEU)	An IBM utility used to write and edit source programs on-line.
Source Program	Programs written in a programming language such as RPG III, COBOL, or FORTRAN. Source programs are readable and must be compiled before they can be executed.
Special Authority	Systemwide authorities not related to a particular object. Special authorities include Job Control Rights and System Save Rights.
Split Password Program	A program included in the CPF programmers guide that an installation can use to control password conventions such as requiring that passwords be changed periodically.
Subsystems	An operating environment used by the operating system to control processing. An example is the spooling subsystem that controls the submission and printing of jobs.

GLOSSARY

System Authority	Systemwide access authority; composed of Special Authority and Object Authority.
System Save Rights	A Special Authority, which is systemwide rather than specific to certain objects, that specifically allows a user to save and restore all objects on the system.
User	Anyone using the system. A user may be a data processing or nondata processing "user" of the system.
User Profile	The parameters describing an individual or group of individuals and the list of objects the user has explicit access to.
Utility	A general purpose program that performs a routine function. Examples include the Data File Utility and the Source Entry Utility.
Work Management	The component of the CPF operating system that directs job input, execution, and output.
Workstation	A computer terminal.

Bibliography

To obtain more information about the System/38, you may wish to refer to these IBM manuals:

<u>Order Number</u>	<u>Title</u>
GC21-7728	<u>IBM System/38 Introduction</u>
GC21-7729	<u>IBM System/38 Control Program Facility Concepts Manual</u>
SC21-7730	<u>IBM System/38 Control Program Facility Programmer's Guide, Chapter 21, Security Aids</u>
SC21-7731	<u>IBM System/38 Control Language Reference Manual</u>
SC21-7714	<u>IBM System/38 Data File Utility Reference Manual and User's Guide</u>
SC21-7724	<u>IBM System/38 Query Utility Reference Manual and User's Guide</u>
SC21-7735	<u>IBM System/38 Operator's Guide</u>

Control Features of the IBM System/38
Ernst & Whinney

Match the major components of control program facility with a description.

- | | |
|--------------------------|--|
| ___ 1. Control language | A. Framework for managing the system and all work performed on the system. |
| ___ 2. Object management | B. Used to create, delete, and modify data objects |
| ___ 3. Work (Job) Mgt. | C. Used to create, delete, and administer named items maintained on the system. |
| ___ 4. Data management | D. Instructions that initiate and direct the activities of the operating system. |
5. A DDS does not need to be compiled before it is usable.
A. TRUE B. FALSE
6. Which is not a type of database file that utilizes DDS?
A. Logical files C. Index files
B. Physical files D. Field reference files
7. A source program using a database file in DDS needs to describe the database fields such as in the Data Division in a COBOL program.
A. TRUE B. FALSE
8. Which log can be reviewed for unauthorized use of compilers, unauthorized access attempts, and any other unusual activities?
A. Job log C. Security log
B. System log D. History log
9. Which utility is not part of the IBM's interactive data base utilities (IDU).
A. Display Information Aid (DIF)
B. Source Entry Utility (SEU)
C. Data File Utility (DFU)
D. Query Utility (QRY)
E. Screen Design Aid (SDA)

Match the utility with a description.

- | | |
|-------------|---|
| ___ 10. DIF | A. Allows users to create and maintain source code on-line. |
| ___ 11. SEU | B. Allows users to interactively specify criteria for the extraction, summarization, and presentation of information from a database. |
| ___ 12. DFU | C. Allows the user to interactively design, create, maintain, and test work station display formats and menus for application. |
| ___ 13. QRY | D. Allows the user to quickly create programs that will add to, delete from, create, read, move, or modify data files without using other programming procedures. |
| ___ 14. SDA | E. Programmer productivity aid. |

15. Which is not an object right?

- A. Administration rights
- B. Object existence rights
- C. Object management rights
- D. Operational rights

16. Which is not a data right?

- A. Read
- B. Update
- C. Move
- D. Add
- E. Delete

Match the object right with its definition.

- 17. Object existence rights A. User can use the object and view the object's description.
- 18. Object management rights B. User can delete, save, restore, and transfer ownership of the object.
- 19. Operational rights C. User can move, rename, grant authority to revoke authority from, and modify the attributes of the object.
- 20. The owner of an object has object rights but not data rights.
 - A. TRUE
 - B. FALSE
- 21. All IBM-supplied user profiles are installed with the same set of passwords printed in the system/38 system documentation manuals. Therefore, anyone with a system/38 manual or prior experience with a system/38 may know the passwords.
 - A. TRUE
 - B. FALSE
- 22. A user must have rights to the library containing an object before the object can be accessed. For example, if a database within a library is defined as PUBAUT(*ALL) but the library is defined as PUBAUT (*NONE) only those users with explicit rights to the library can access the database.
 - A. TRUE
 - B. FALSE
- 23. The system/38 console is equipped with external rotary switches that can be set to allow anyone to display or alter main storage. These switches are on the exterior and can be locked.
 - A. TRUE
 - B. FALSE
- 24. The security officer user profile is authorized to use all work stations by default.
 - A. TRUE
 - B. FALSE
- 25. A user profile object name and a password are required to sign-on to the system.
 - A. TRUE
 - B. FALSE

